FIG. 1

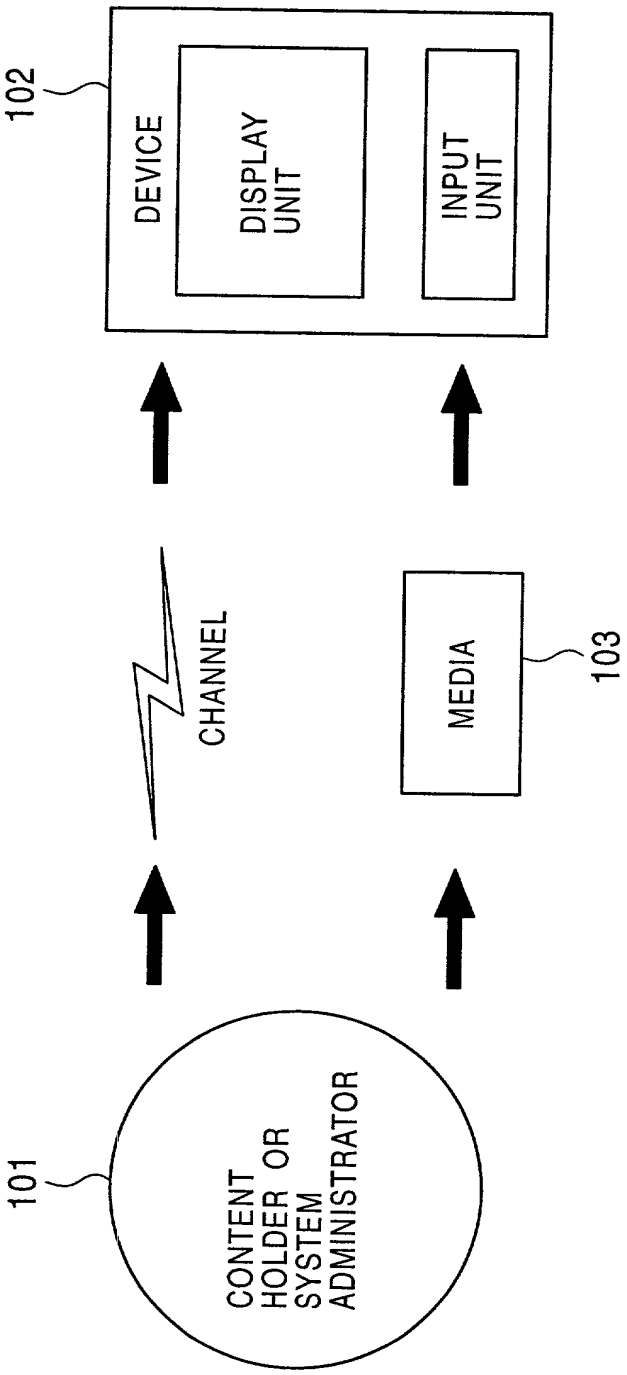

102
DEVICE
DISPLAY UNIT
INPUT UNIT
CHANNEL
MEDIA
103
101
CONTENT HOLDER OR SYSTEM ADMINISTRATOR

FIG. 2

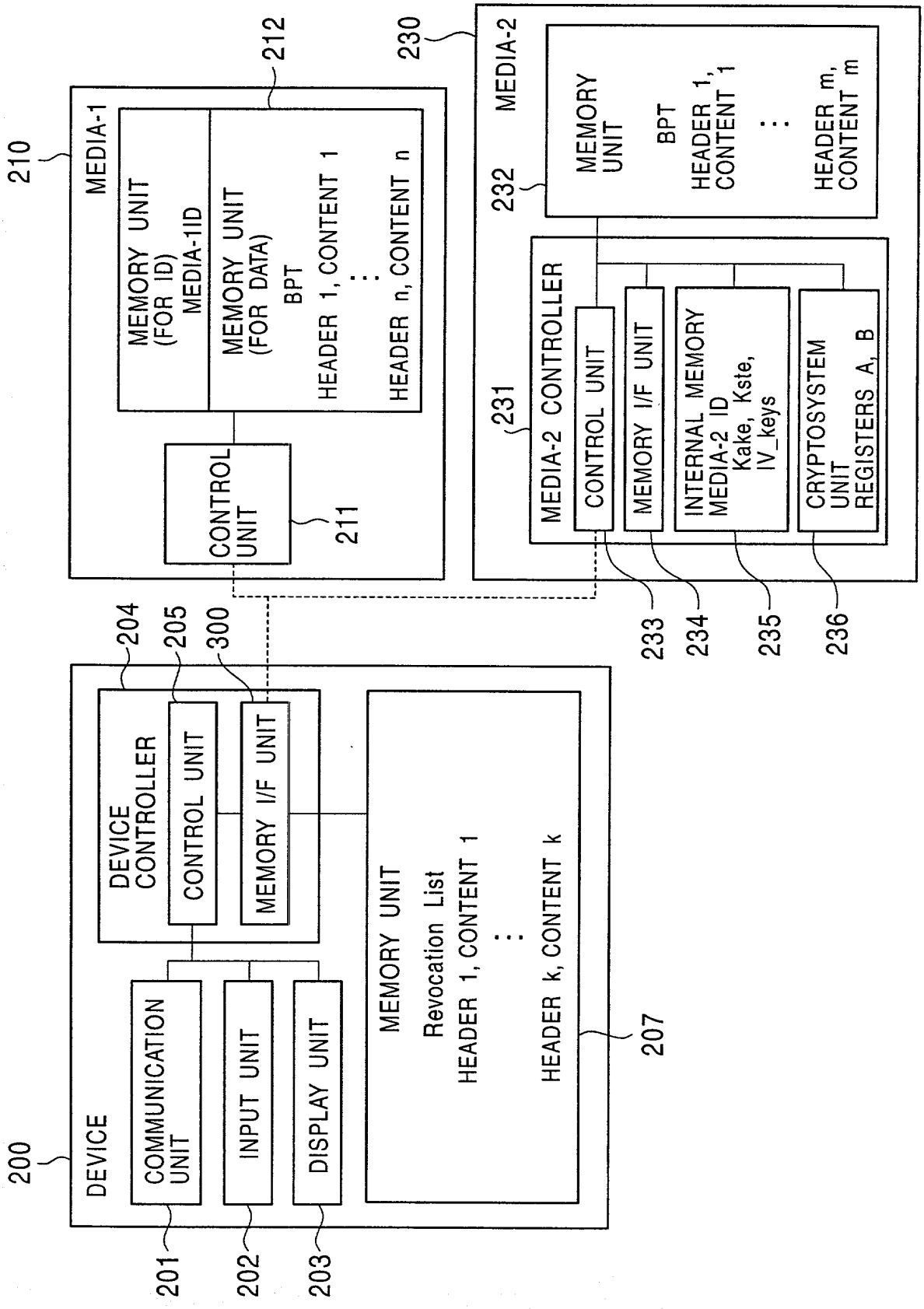
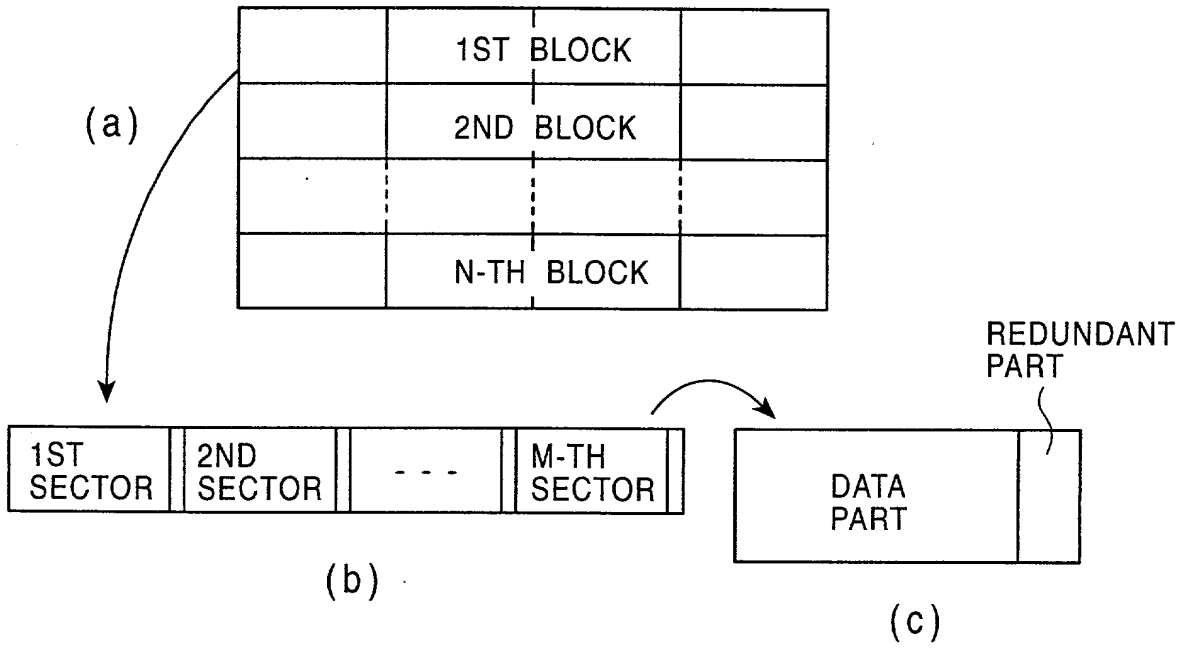

200
DEVICE
COMMUNICATION UNIT
INPUT UNIT
DISPLAY UNIT
201
202
203
DEVICE CONTROLLER
CONTROL UNIT
MEMORY I/F UNIT
204
205
300
MEMORY UNIT
Revocation List
HEADER 1, CONTENT 1
HEADER k, CONTENT k
207
210
MEDIA-1
MEMORY UNIT (FOR ID)
MEDIA-1ID
MEMORY UNIT (FOR DATA)
BPT
HEADER 1, CONTENT 1
HEADER n, CONTENT n
212
CONTROL UNIT
211
230
MEDIA-2
232
MEMORY UNIT
BPT
HEADER 1, CONTENT 1
HEADER m, CONTENT m
231
MEDIA-2 CONTROLLER
CONTROL UNIT
MEMORY I/F UNIT
INTERNAL MEMORY
MEDIA-2 ID
Kake, Kste, IV_keys
CRYPTOSYSTEM UNIT
REGISTERS A, B
233
234
235
236

FIG. 3


1ST BLOCK
2ND BLOCK
N-TH BLOCK
(a)
1ST SECTOR
2ND SECTOR
- - -
M-TH SECTOR
(b)
REDUNDANT PART
DATA PART
(c)

FOR OFFICIAL USE ONLY

FIG. 4

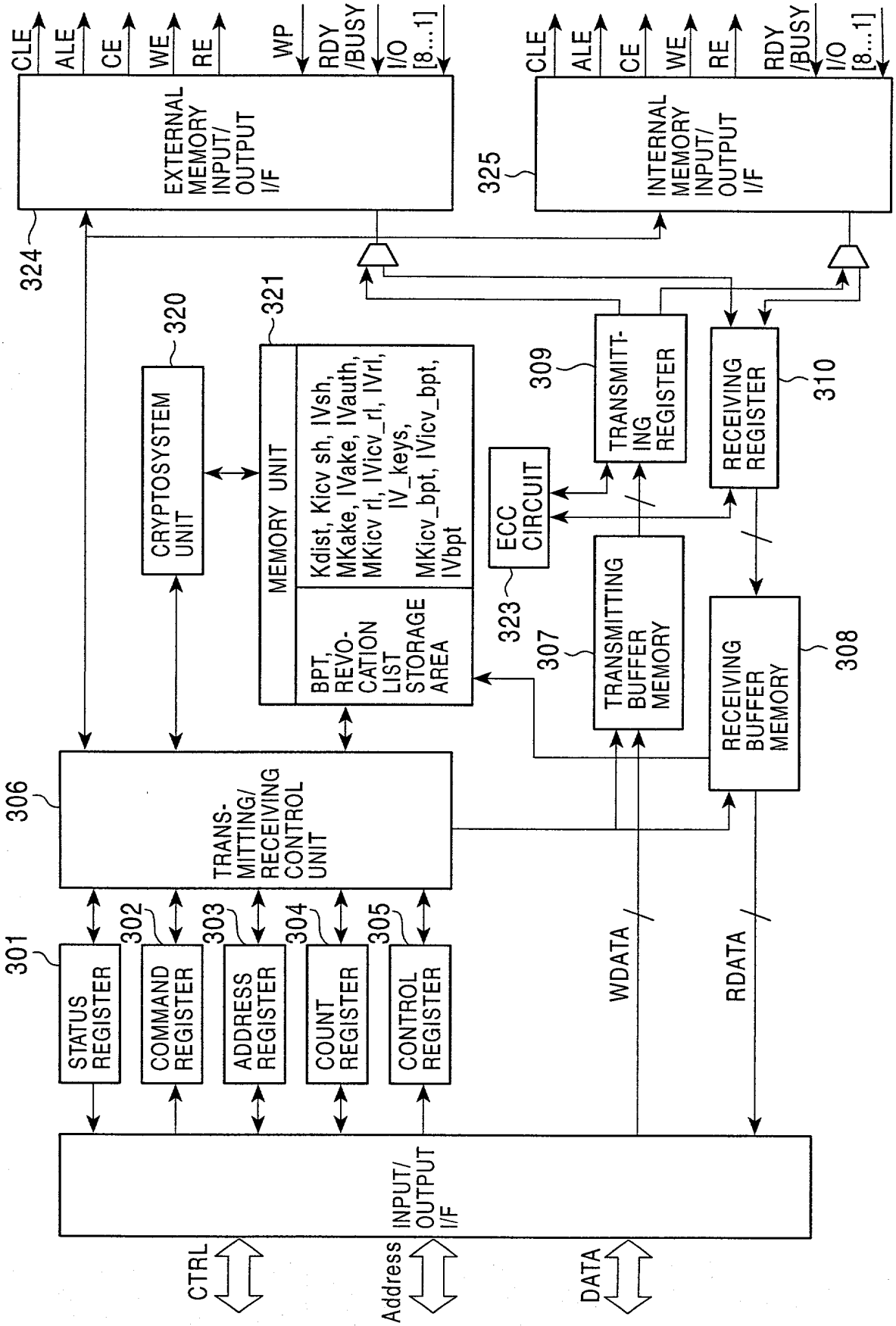


FIG. 5

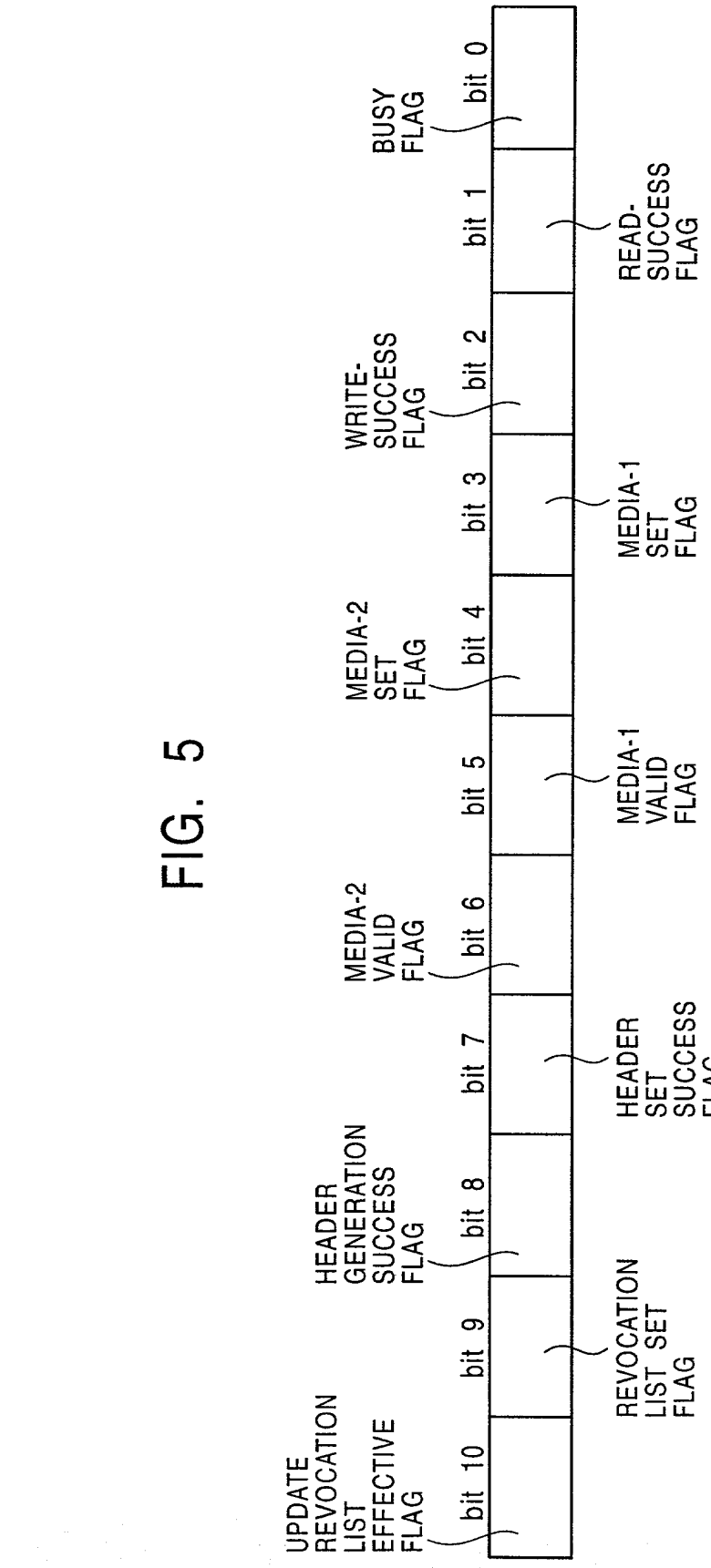

bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
bit 8
bit 9
bit 10
BUSY FLAG
READ-SUCCESS FLAG
WRITE-SUCCESS FLAG
MEDIA-1 SET FLAG
MEDIA-2 SET FLAG
MEDIA-1 VALID FLAG
MEDIA-2 VALID FLAG
HEADER SET SUCCESS FLAG
HEADER GENERATION SUCCESS FLAG
REVOCATION LIST SET FLAG
UPDATE REVOCATION LIST EFFECTIVE FLAG

FIG. 6A

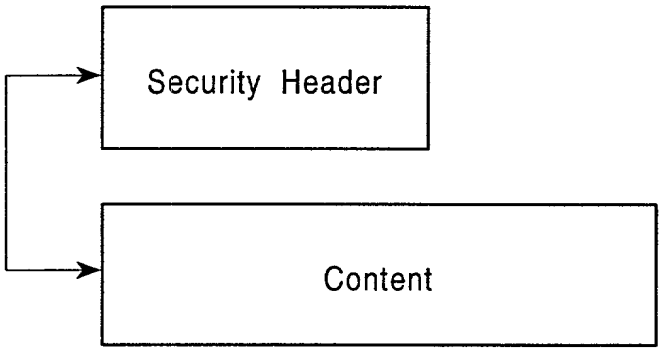
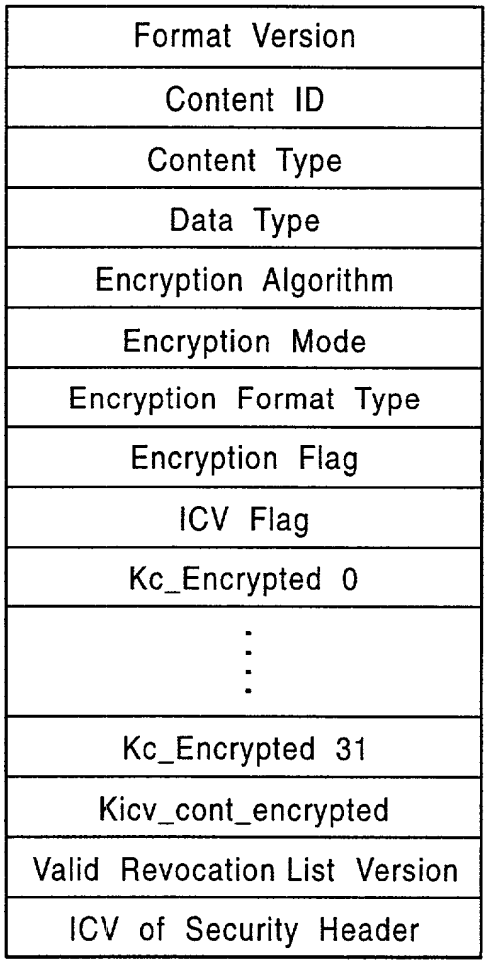


FIG. 6B

Block #1	Security Header1
Block #2	Content 1
Block #3	Content 1
Block #4	Security Header 2
Block #5	Content 2
Block #6	Security Header 3
Block #7	Content 3
⋮	⋮

FIG. 7



EP 1 615 369 A1

FIG. 8

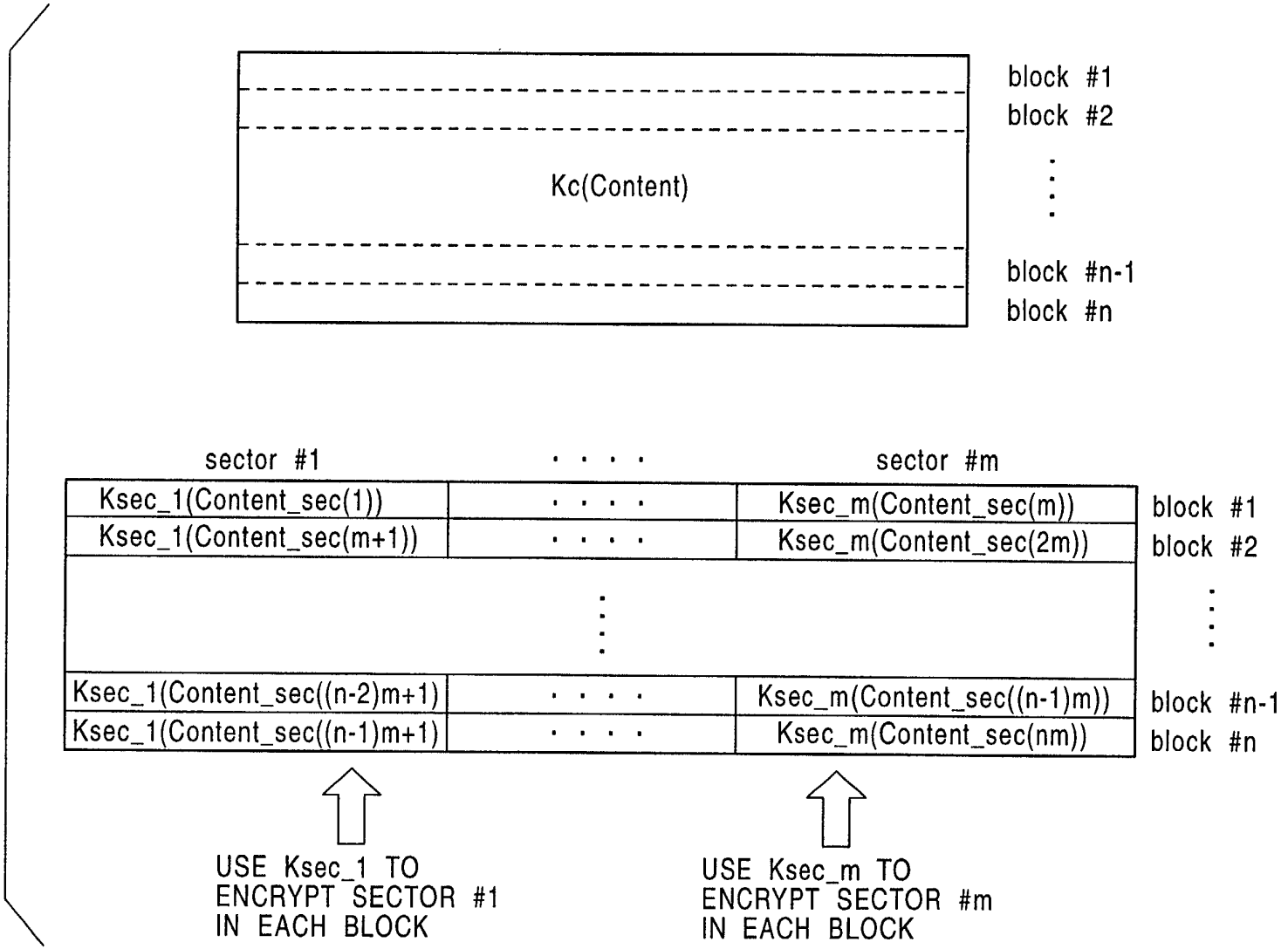

Kc(Content)
block #1
block #2
block #n-1
block #n
sector #1
sector #m
Ksec_1(Content_sec(1))
Ksec_m(Content_sec(m))
Ksec_1(Content_sec(m+1))
Ksec_m(Content_sec(2m))
Ksec_1(Content_sec((n-2)m+1))
Ksec_m(Content_sec((n-1)m))
Ksec_1(Content_sec((n-1)m+1))
Ksec_m(Content_sec(nm))
block #1
block #2
block #n-1
block #n
USE Ksec_1 TO ENCRYPT SECTOR #1 IN EACH BLOCK
USE Ksec_m TO ENCRYPT SECTOR #m IN EACH BLOCK

FIG. 9

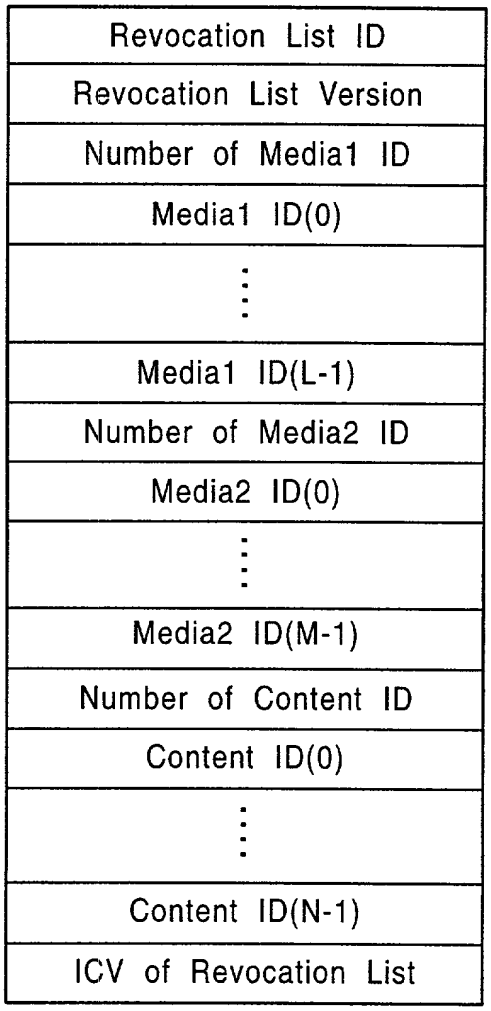

Revocation List ID
Revocation List Version
Number of Media1 ID
Media1 ID(0)
Media1 ID(L-1)
Number of Media2 ID
Media2 ID(0)
Media2 ID(M-1)
Number of Content ID
Content ID(0)
Content ID(N-1)
ICV of Revocation List

FIG. 10

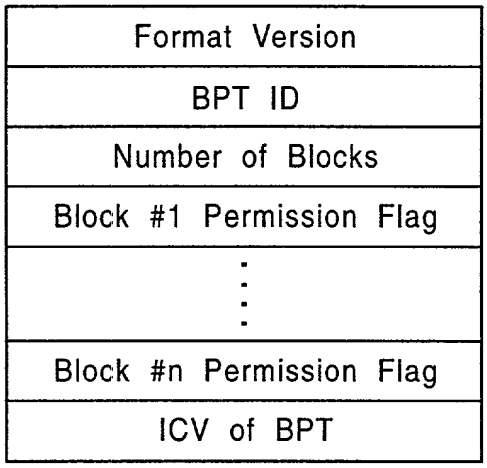

Format Version
BPT ID
Number of Blocks
Block #1 Permission Flag
Block #n Permission Flag
ICV of BPT

FIG. 11

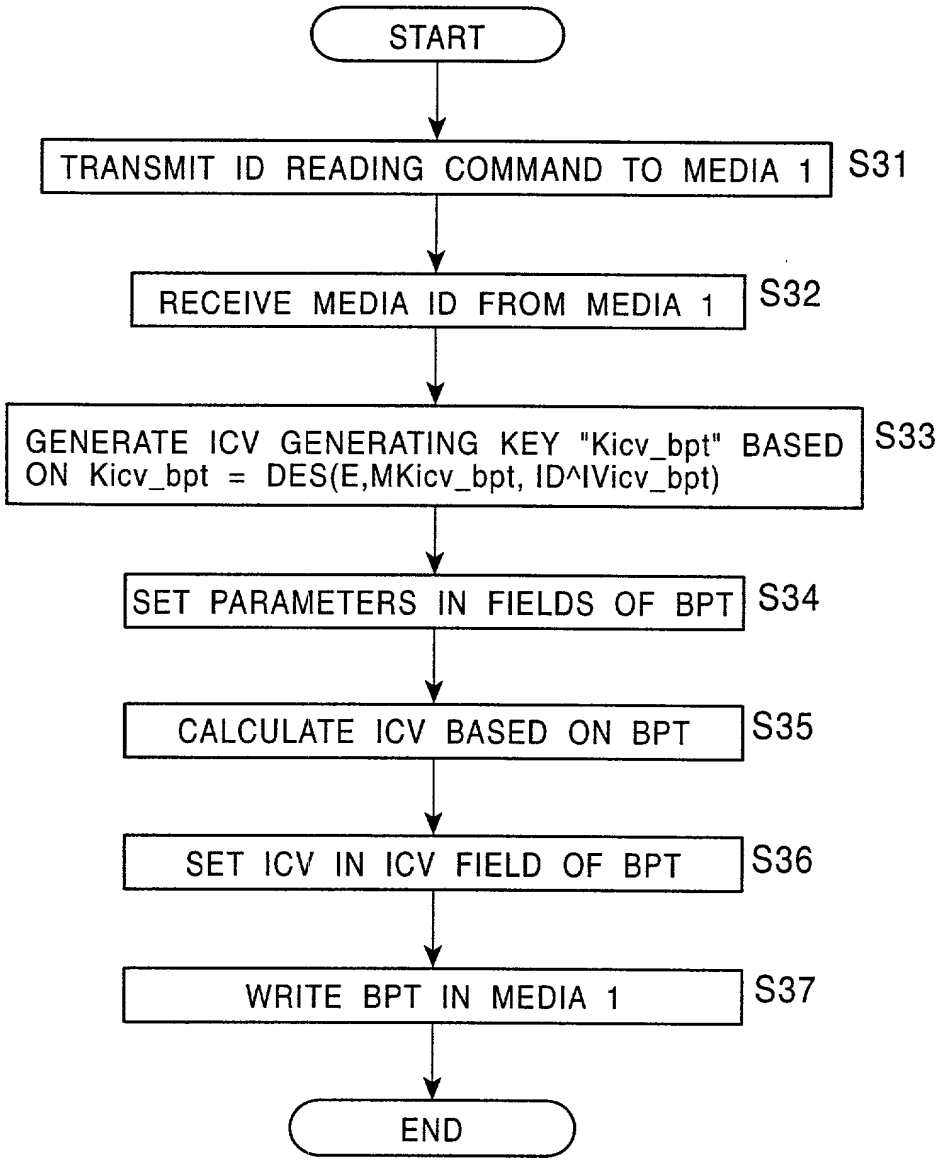

START
TRANSMIT ID READING COMMAND TO MEDIA 1
S31
RECEIVE MEDIA ID FROM MEDIA 1
S32
GENERATE ICV GENERATING KEY "Kicv_bpt" BASED ON Kicv_bpt = DES(E,MKicv_bpt, ID^IVicv_bpt)
S33
SET PARAMETERS IN FIELDS OF BPT
S34
CALCULATE ICV BASED ON BPT
S35
SET ICV IN ICV FIELD OF BPT
S36
WRITE BPT IN MEDIA 1
S37
END

FIG. 12

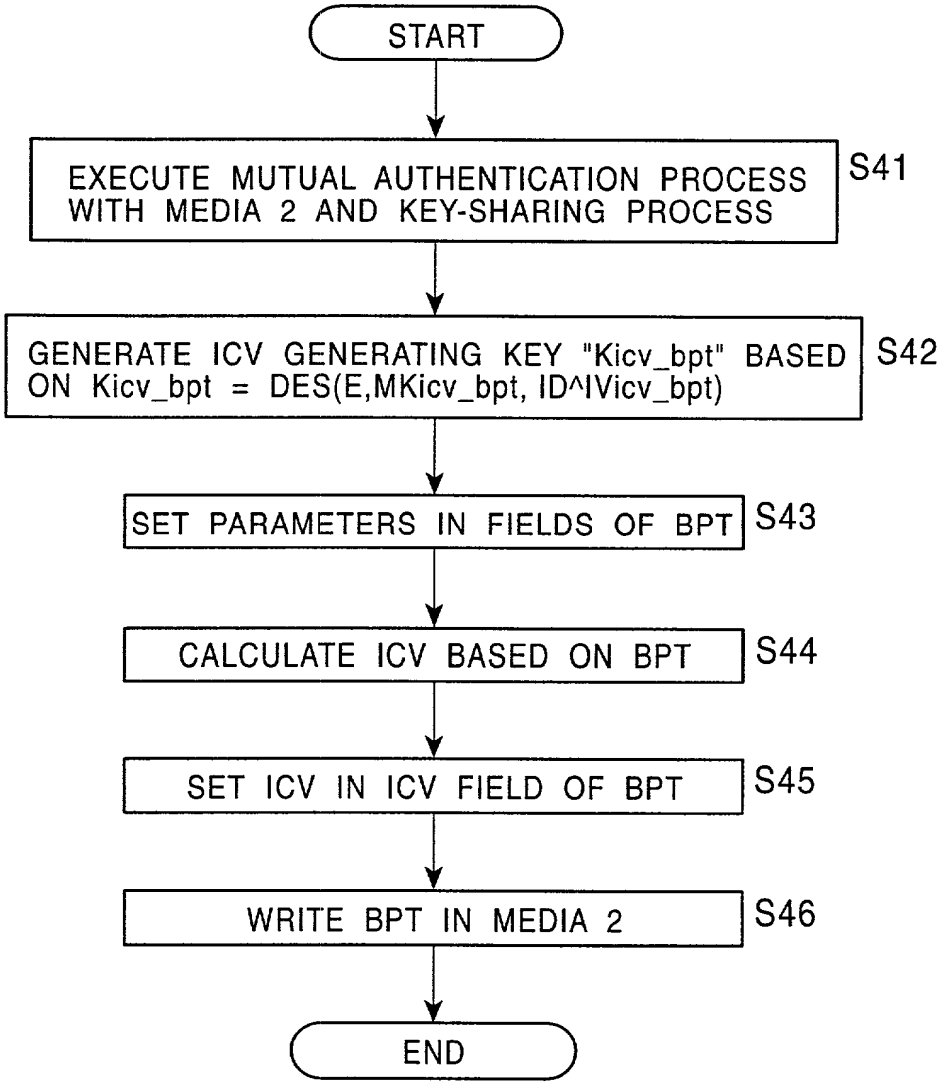

START
EXECUTE MUTUAL AUTHENTICATION PROCESS WITH MEDIA 2 AND KEY-SHARING PROCESS
S41
GENERATE ICV GENERATING KEY "Kicv_bpt" BASED ON Kicv_bpt = DES(E,MKicv_bpt, ID^IVicv_bpt)
S42
SET PARAMETERS IN FIELDS OF BPT
S43
CALCULATE ICV BASED ON BPT
S44
SET ICV IN ICV FIELD OF BPT
S45
WRITE BPT IN MEDIA 2
S46
END

FIG. 13

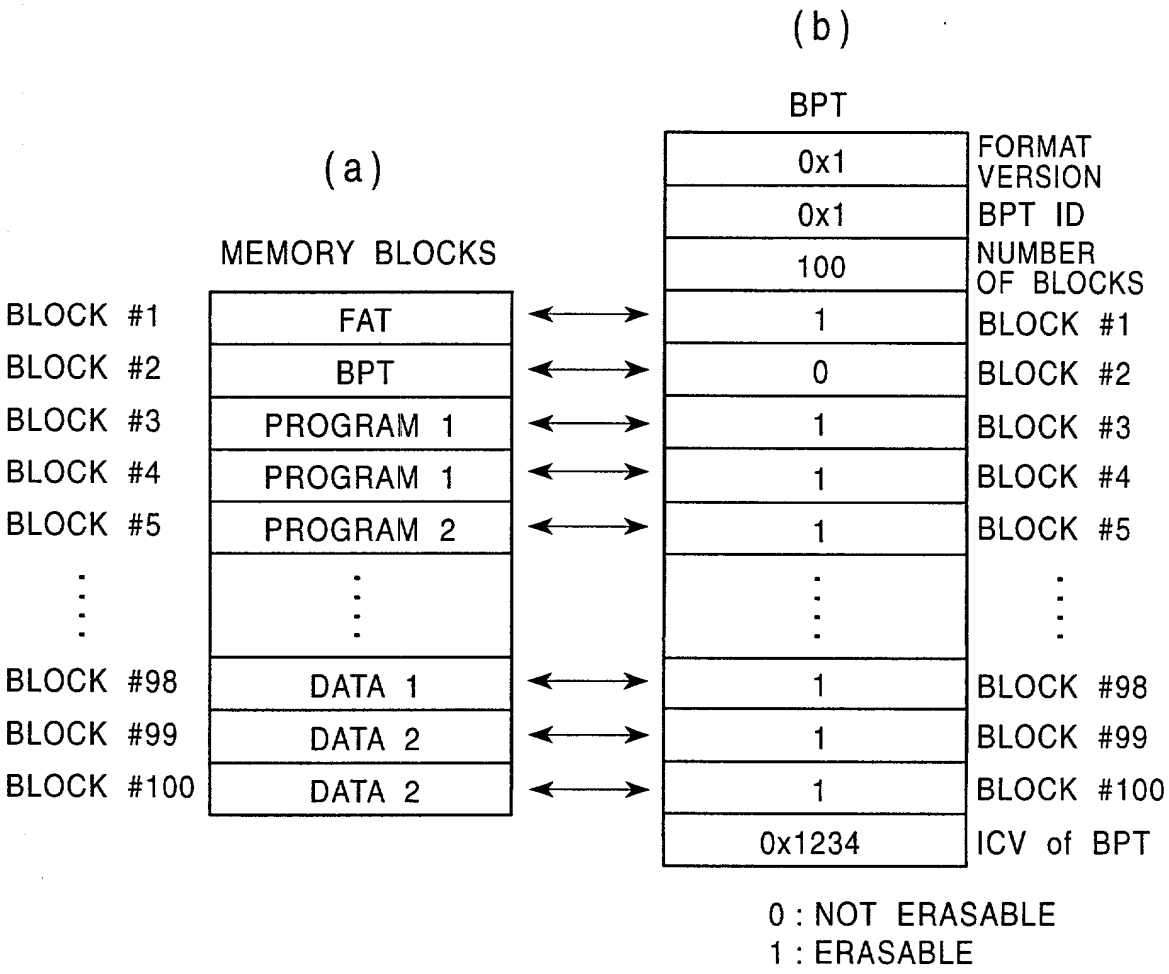

(a)
MEMORY BLOCKS
BLOCK #1
BLOCK #2
BLOCK #3
BLOCK #4
BLOCK #5
BLOCK #98
BLOCK #99
BLOCK #100
FAT
BPT
PROGRAM 1
PROGRAM 1
PROGRAM 2
DATA 1
DATA 2
DATA 2
(b)
BPT
0x1
FORMAT VERSION
0x1
BPT ID
100
NUMBER OF BLOCKS
1
BLOCK #1
0
BLOCK #2
1
BLOCK #3
1
BLOCK #4
1
BLOCK #5
1
BLOCK #98
1
BLOCK #99
1
BLOCK #100
0x1234
ICV of BPT
0 : NOT ERASABLE
1 : ERASABLE

FIG. 14

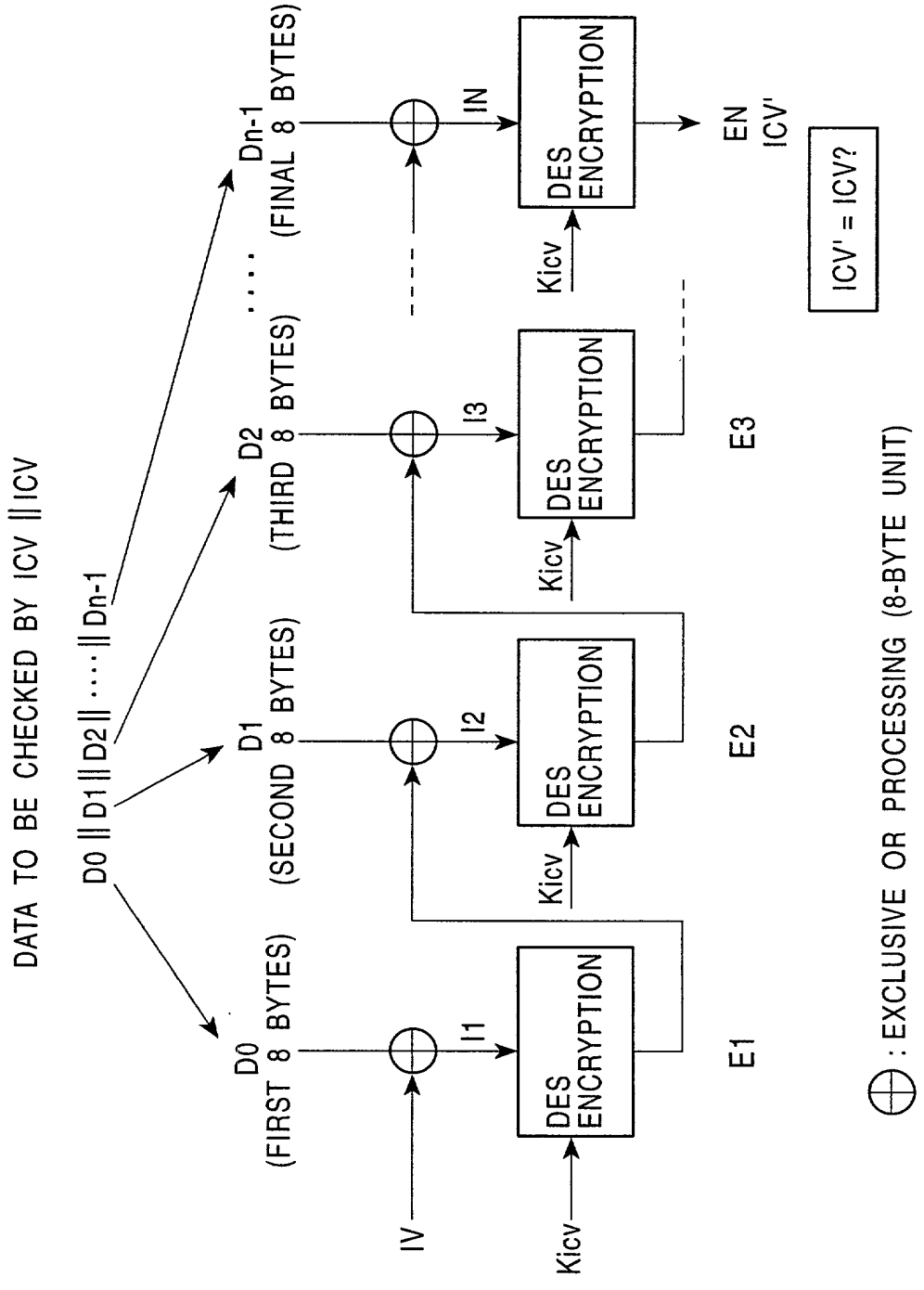

DATA TO BE CHECKED BY ICV || ICV
D0 || D1 || D2 || .... || Dn-1
D0 (FIRST 8 BYTES)
D1 (SECOND 8 BYTES)
D2 (THIRD 8 BYTES)
.....
Dn-1 (FINAL 8 BYTES)
IV
Kicv
I1
I2
I3
IN
DES ENCRYPTION
E1
E2
E3
EN
ICV'
ICV' = ICV?
⊕ : EXCLUSIVE OR PROCESSING (8-BYTE UNIT)

FIG. 15

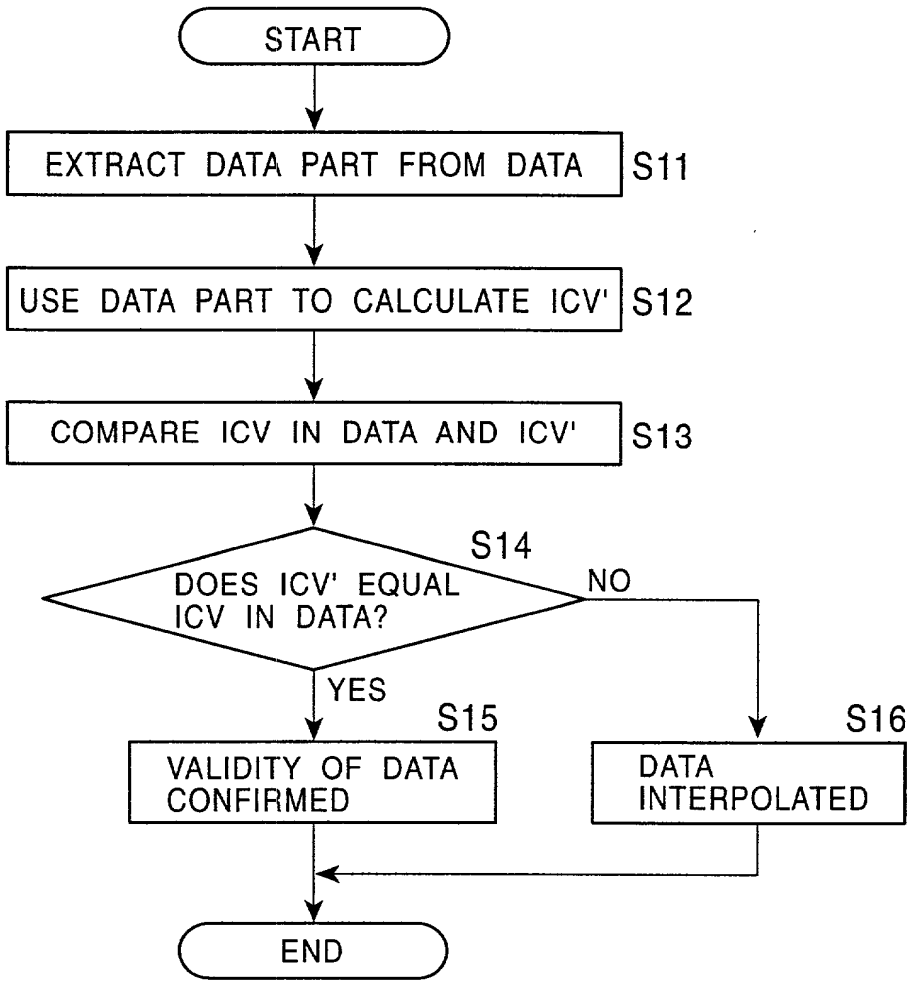

START
EXTRACT DATA PART FROM DATA
S11
USE DATA PART TO CALCULATE ICV'
S12
COMPARE ICV IN DATA AND ICV'
S13
S14
DOES ICV' EQUAL ICV IN DATA?
NO
YES
S15
VALIDITY OF DATA CONFIRMED
S16
DATA INTERPOLATED
END

FIG. 16

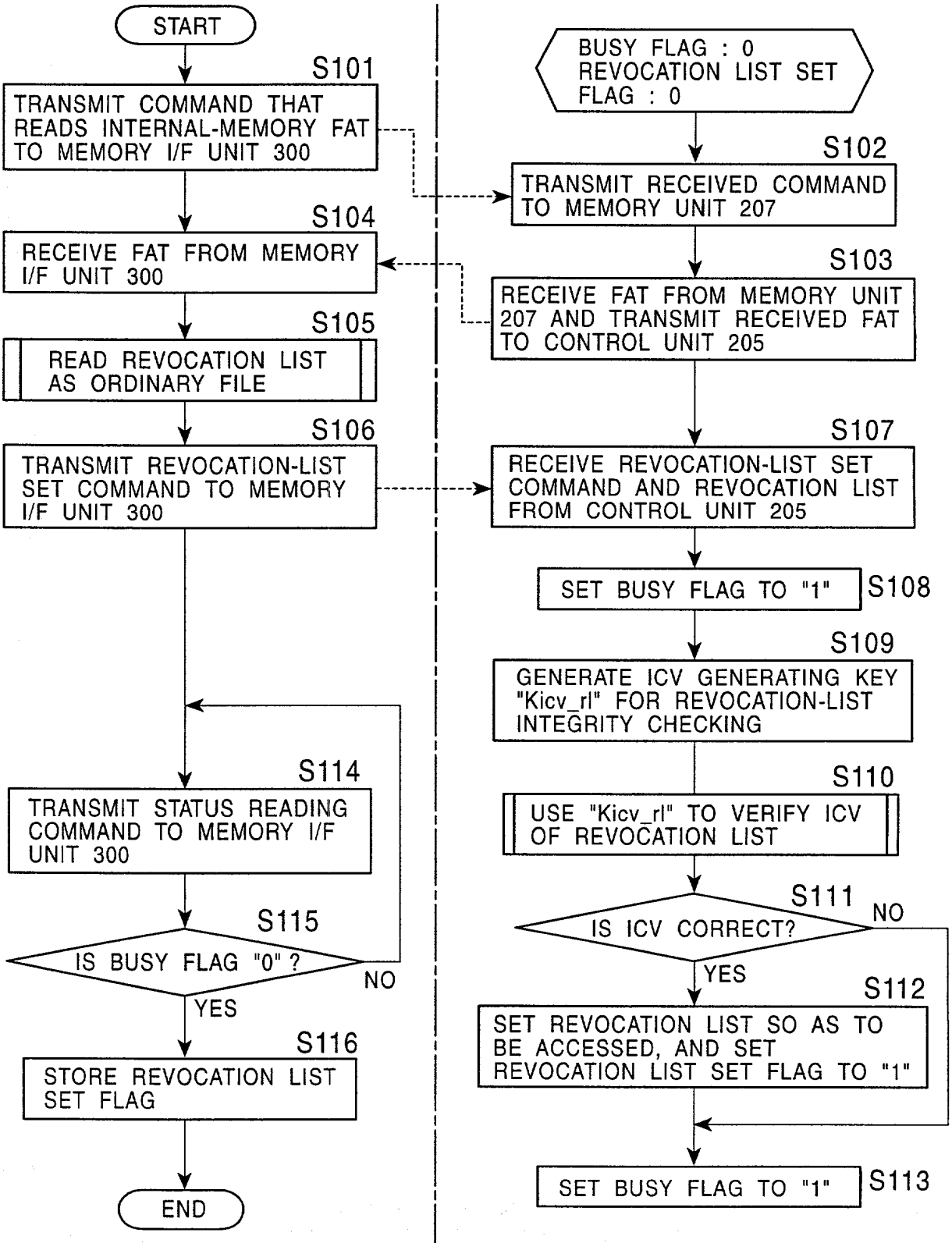

START
S101
TRANSMIT COMMAND THAT READS INTERNAL-MEMORY FAT TO MEMORY I/F UNIT 300
S104
RECEIVE FAT FROM MEMORY I/F UNIT 300
S105
READ REVOCATION LIST AS ORDINARY FILE
S106
TRANSMIT REVOCATION-LIST SET COMMAND TO MEMORY I/F UNIT 300
S114
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S115
IS BUSY FLAG "0" ?
NO
YES
S116
STORE REVOCATION LIST SET FLAG
END
BUSY FLAG : 0
REVOCATION LIST SET FLAG : 0
S102
TRANSMIT RECEIVED COMMAND TO MEMORY UNIT 207
S103
RECEIVE FAT FROM MEMORY UNIT 207 AND TRANSMIT RECEIVED FAT TO CONTROL UNIT 205
S107
RECEIVE REVOCATION-LIST SET COMMAND AND REVOCATION LIST FROM CONTROL UNIT 205
SET BUSY FLAG TO "1"
S108
S109
GENERATE ICV GENERATING KEY "Kicv_rl" FOR REVOCATION-LIST INTEGRITY CHECKING
S110
USE "Kicv_rl" TO VERIFY ICV OF REVOCATION LIST
S111
IS ICV CORRECT?
NO
YES
S112
SET REVOCATION LIST SO AS TO BE ACCESSED, AND SET REVOCATION LIST SET FLAG TO "1"
SET BUSY FLAG TO "1"
S113

FIG. 17

DIRECTORY	FILE NAME	STORAGE SECTORS
/	A.h	1 — 10
/	A.cont	21 — 100
/dir_a	B.h	101 — 110
/dir_a	B.cont	111 — 350
/dir_a/dir_x	C	401 — 450
/dir_b	D	501 — 580
/dir_c	E.h	601 — 610
⋮	⋮	⋮
/dir_c	Z.cont	5001 — 5340

FIG. 18

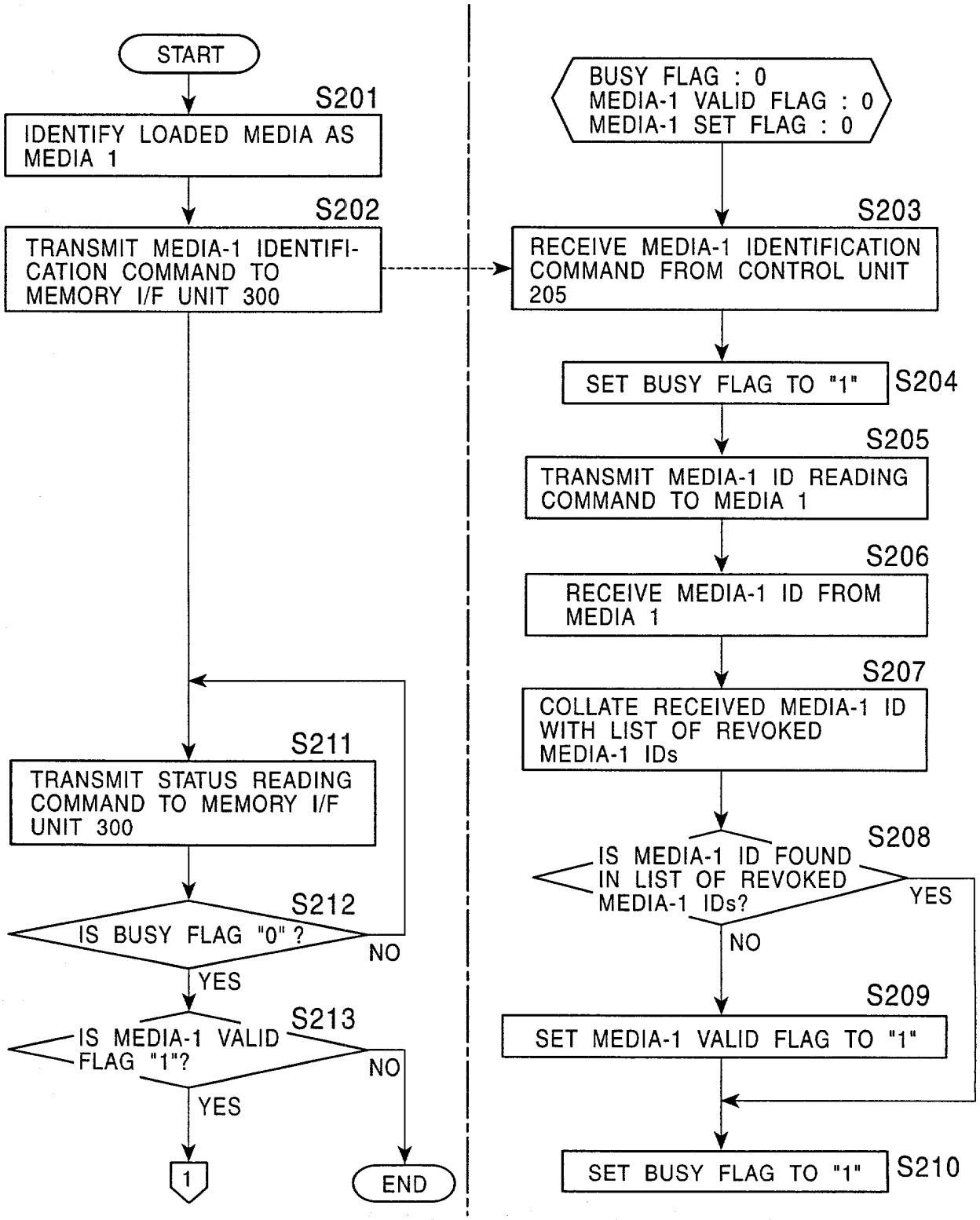

START
S201
IDENTIFY LOADED MEDIA AS MEDIA 1
S202
TRANSMIT MEDIA-1 IDENTIFICATION COMMAND TO MEMORY I/F UNIT 300
BUSY FLAG : 0
MEDIA-1 VALID FLAG : 0
MEDIA-1 SET FLAG : 0
S203
RECEIVE MEDIA-1 IDENTIFICATION COMMAND FROM CONTROL UNIT 205
SET BUSY FLAG TO "1"
S204
S205
TRANSMIT MEDIA-1 ID READING COMMAND TO MEDIA 1
S206
RECEIVE MEDIA-1 ID FROM MEDIA 1
S207
COLLATE RECEIVED MEDIA-1 ID WITH LIST OF REVOKED MEDIA-1 IDs
S208
IS MEDIA-1 ID FOUND IN LIST OF REVOKED MEDIA-1 IDs?
YES
NO
S209
SET MEDIA-1 VALID FLAG TO "1"
SET BUSY FLAG TO "1"
S210
S211
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S212
IS BUSY FLAG "0" ?
NO
YES
S213
IS MEDIA-1 VALID FLAG "1"?
NO
YES
1
END

FIG. 19

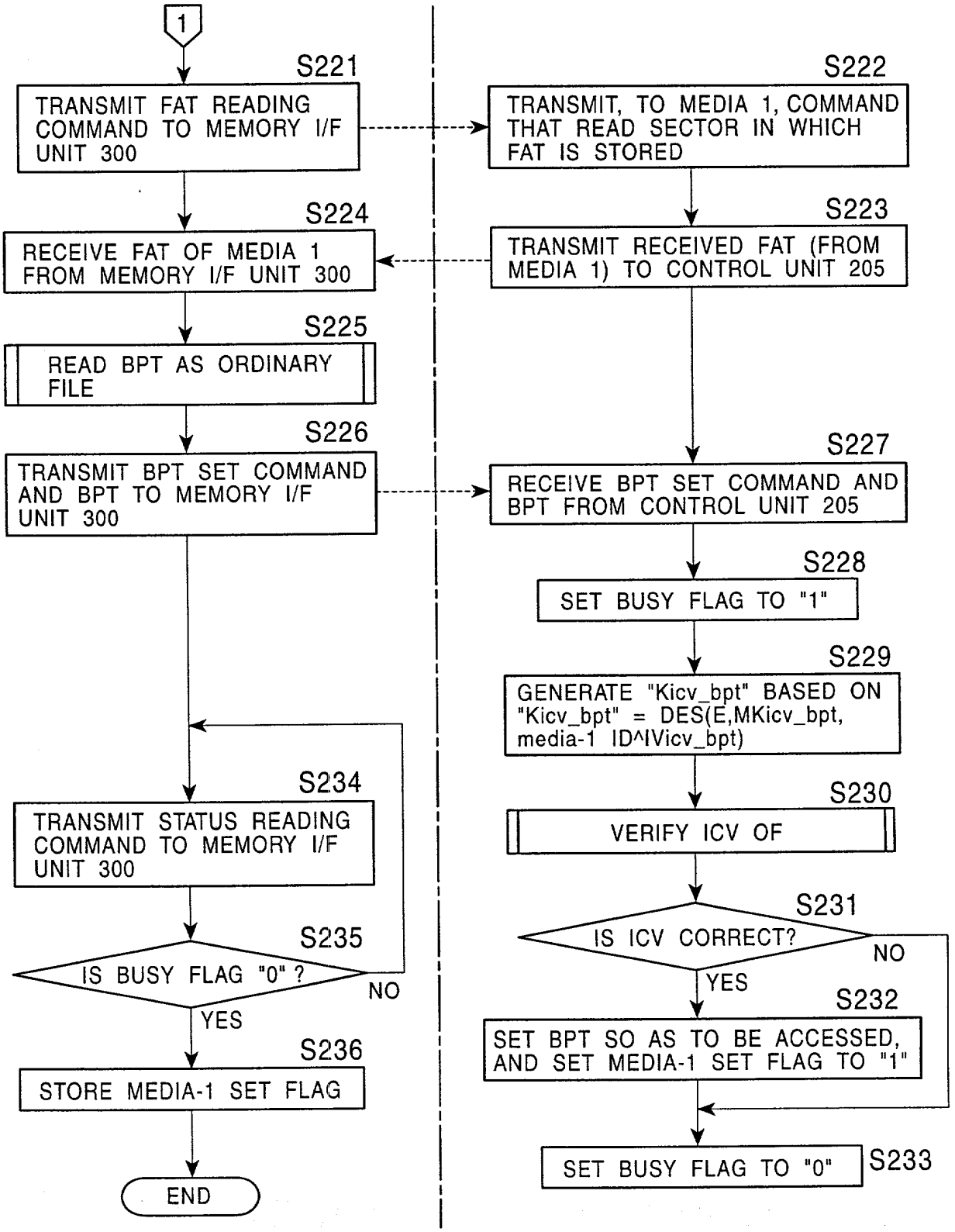

1
S221
TRANSMIT FAT READING COMMAND TO MEMORY I/F UNIT 300
S222
TRANSMIT, TO MEDIA 1, COMMAND THAT READ SECTOR IN WHICH FAT IS STORED
S223
TRANSMIT RECEIVED FAT (FROM MEDIA 1) TO CONTROL UNIT 205
S224
RECEIVE FAT OF MEDIA 1 FROM MEMORY I/F UNIT 300
S225
READ BPT AS ORDINARY FILE
S226
TRANSMIT BPT SET COMMAND AND BPT TO MEMORY I/F UNIT 300
S227
RECEIVE BPT SET COMMAND AND BPT FROM CONTROL UNIT 205
S228
SET BUSY FLAG TO "1"
S229
GENERATE "Kicv_bpt" BASED ON "Kicv_bpt" = DES(E,MKicv_bpt, media-1 ID^IVicv_bpt)
S230
VERIFY ICV OF
S231
IS ICV CORRECT?
NO
YES
S232
SET BPT SO AS TO BE ACCESSED, AND SET MEDIA-1 SET FLAG TO "1"
S233
SET BUSY FLAG TO "0"
S234
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S235
IS BUSY FLAG "0" ?
NO
YES
S236
STORE MEDIA-1 SET FLAG
END

FIG. 20

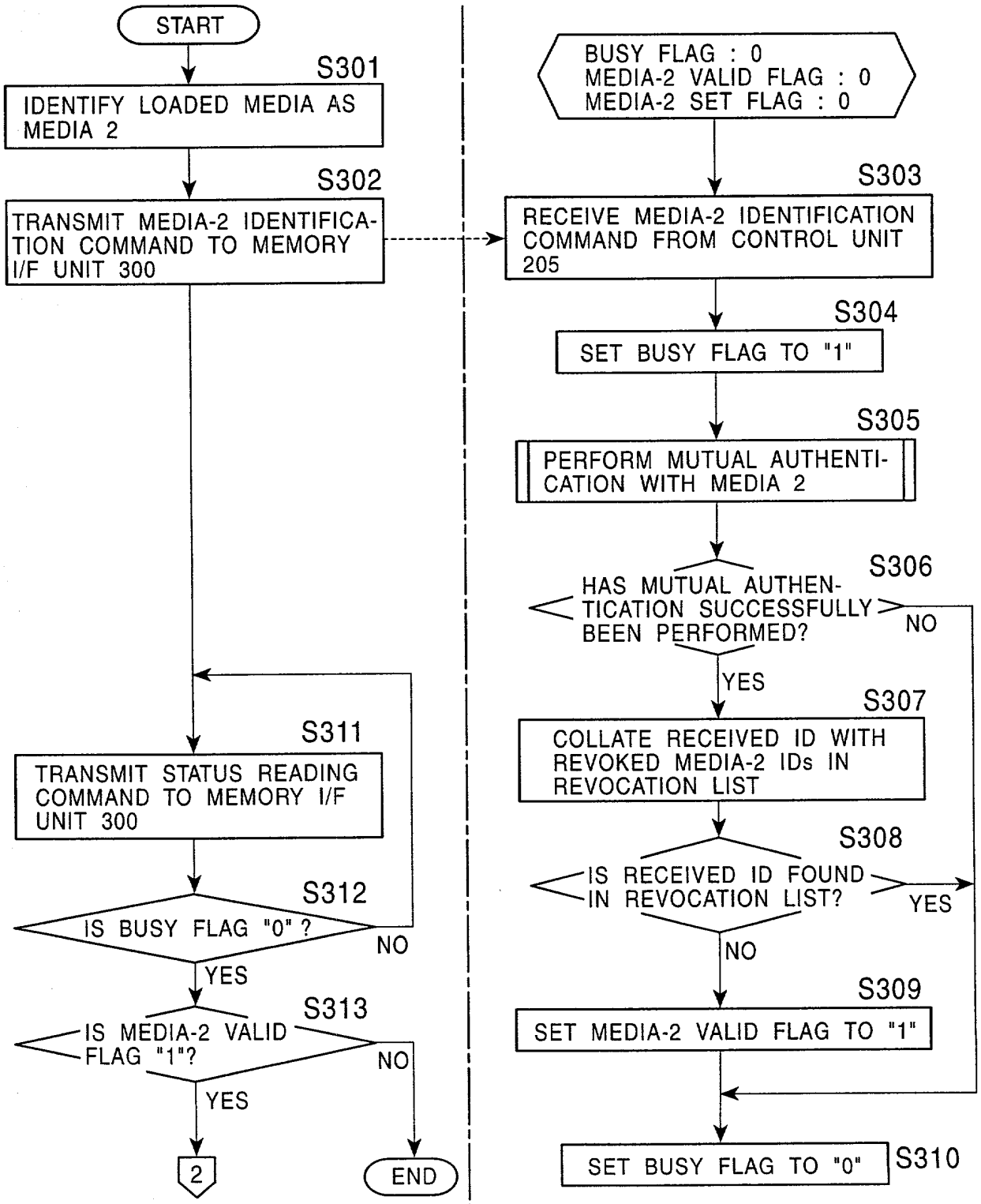
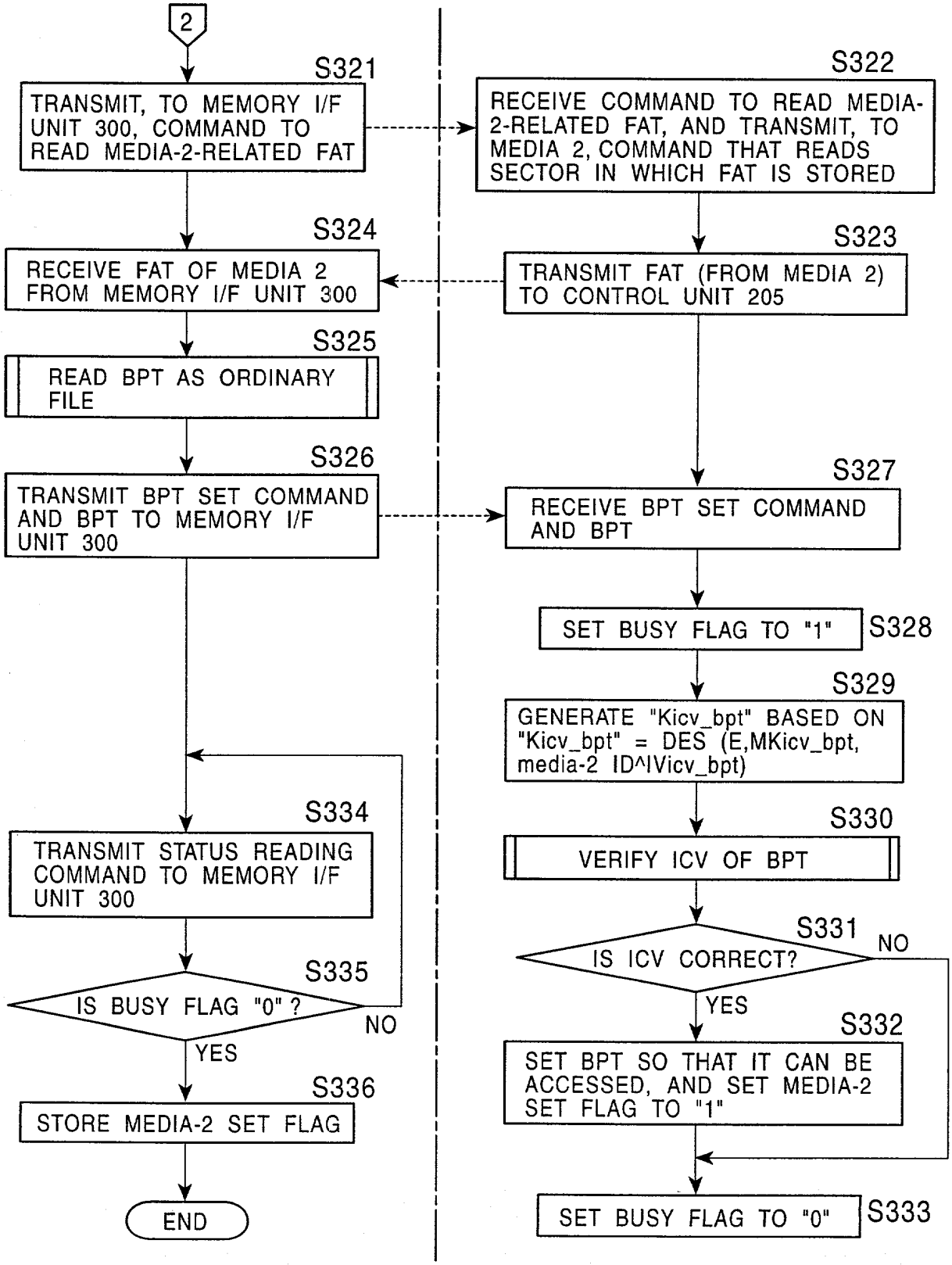

START
S301
IDENTIFY LOADED MEDIA AS MEDIA 2
S302
TRANSMIT MEDIA-2 IDENTIFICATION COMMAND TO MEMORY I/F UNIT 300
S311
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S312
IS BUSY FLAG "0" ?
NO
YES
S313
IS MEDIA-2 VALID FLAG "1"?
NO
YES
2
END
BUSY FLAG : 0
MEDIA-2 VALID FLAG : 0
MEDIA-2 SET FLAG : 0
S303
RECEIVE MEDIA-2 IDENTIFICATION COMMAND FROM CONTROL UNIT 205
S304
SET BUSY FLAG TO "1"
S305
PERFORM MUTUAL AUTHENTICATION WITH MEDIA 2
S306
HAS MUTUAL AUTHENTICATION SUCCESSFULLY BEEN PERFORMED?
NO
YES
S307
COLLATE RECEIVED ID WITH REVOKED MEDIA-2 IDs IN REVOCATION LIST
S308
IS RECEIVED ID FOUND IN REVOCATION LIST?
YES
NO
S309
SET MEDIA-2 VALID FLAG TO "1"
S310
SET BUSY FLAG TO "0"

FIG. 21


2
S321
TRANSMIT, TO MEMORY I/F UNIT 300, COMMAND TO READ MEDIA-2-RELATED FAT
S322
RECEIVE COMMAND TO READ MEDIA-2-RELATED FAT, AND TRANSMIT, TO MEDIA 2, COMMAND THAT READS SECTOR IN WHICH FAT IS STORED
S323
TRANSMIT FAT (FROM MEDIA 2) TO CONTROL UNIT 205
S324
RECEIVE FAT OF MEDIA 2 FROM MEMORY I/F UNIT 300
S325
READ BPT AS ORDINARY FILE
S326
TRANSMIT BPT SET COMMAND AND BPT TO MEMORY I/F UNIT 300
S327
RECEIVE BPT SET COMMAND AND BPT
SET BUSY FLAG TO "1"
S328
S329
GENERATE "Kicv_bpt" BASED ON "Kicv_bpt" = DES (E,MKicv_bpt, media-2 ID^IVicv_bpt)
S330
VERIFY ICV OF BPT
S331
IS ICV CORRECT?
NO
YES
S332
SET BPT SO THAT IT CAN BE ACCESSED, AND SET MEDIA-2 SET FLAG TO "1"
SET BUSY FLAG TO "0"
S333
S334
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S335
IS BUSY FLAG "0" ?
NO
YES
S336
STORE MEDIA-2 SET FLAG
END

EPO - DG 1 - 05.06.2007

FIG. 22

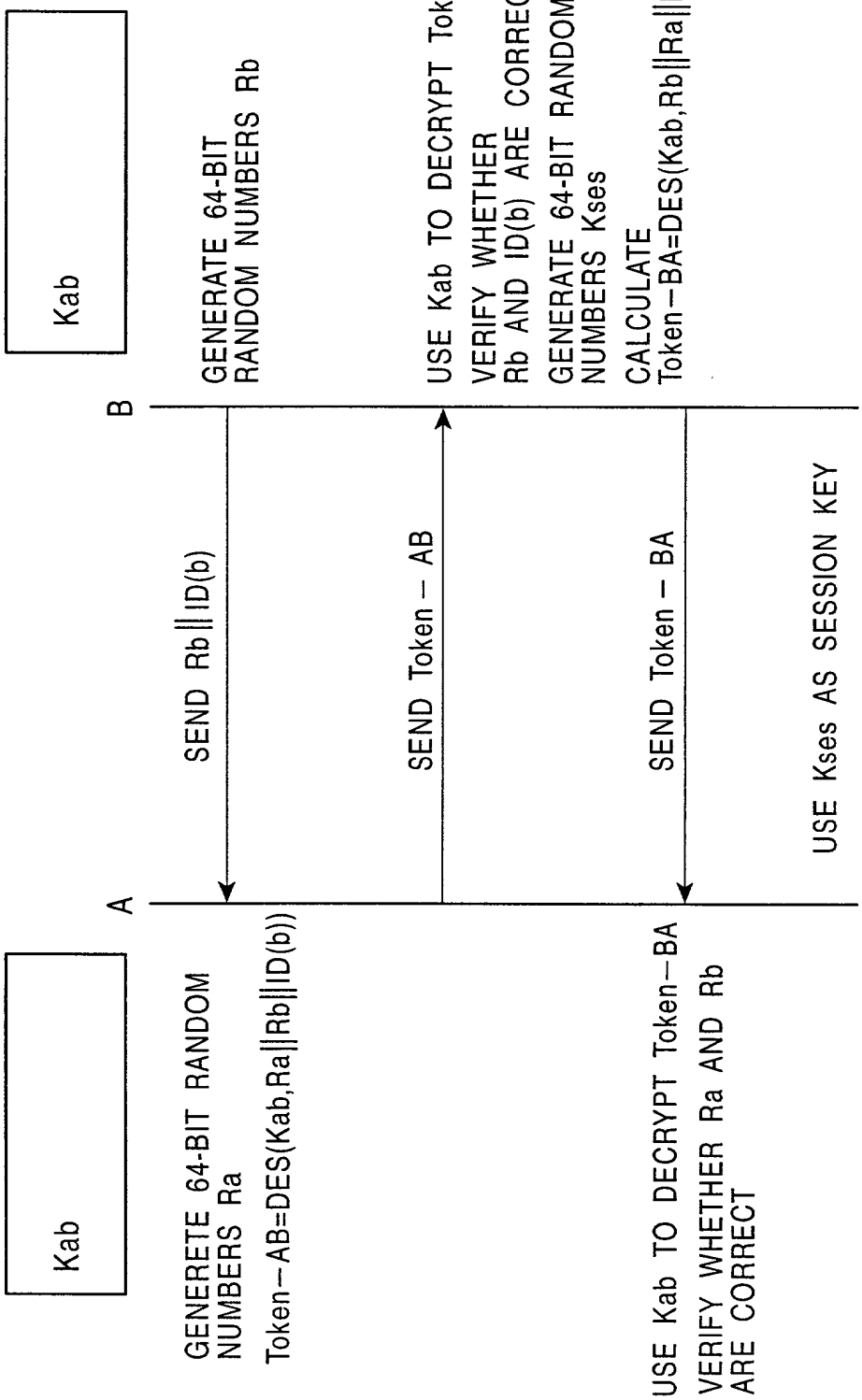


FIG. 23

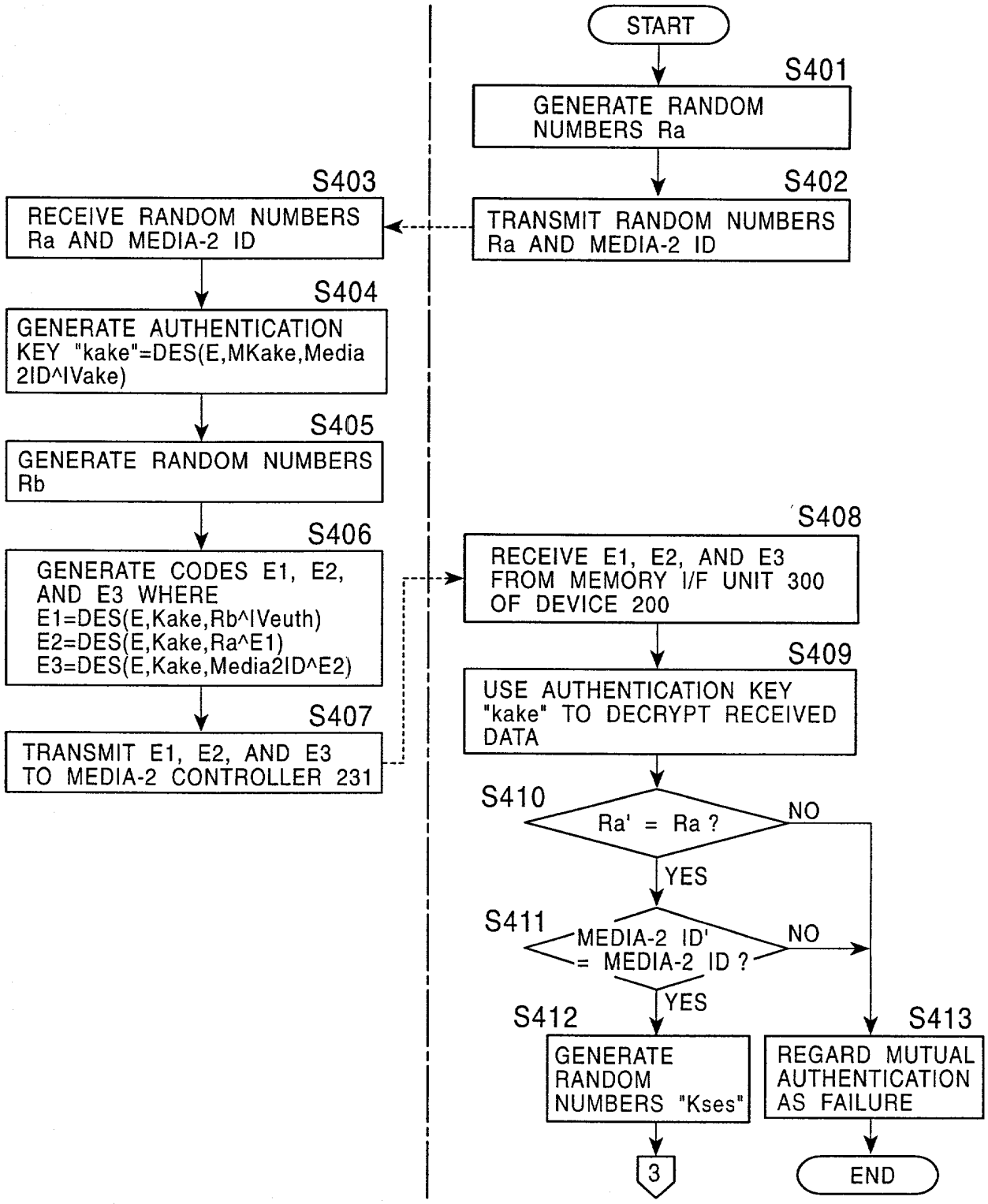

START
S401
GENERATE RANDOM NUMBERS Ra
S402
TRANSMIT RANDOM NUMBERS Ra AND MEDIA-2 ID
S403
RECEIVE RANDOM NUMBERS Ra AND MEDIA-2 ID
S404
GENERATE AUTHENTICATION KEY "kake"=DES(E,MKake,Media 2ID^IVake)
S405
GENERATE RANDOM NUMBERS Rb
S406
GENERATE CODES E1, E2, AND E3 WHERE E1=DES(E,Kake,Rb^IVeuth) E2=DES(E,Kake,Ra^E1) E3=DES(E,Kake,Media2ID^E2)
S407
TRANSMIT E1, E2, AND E3 TO MEDIA-2 CONTROLLER 231
S408
RECEIVE E1, E2, AND E3 FROM MEMORY I/F UNIT 300 OF DEVICE 200
S409
USE AUTHENTICATION KEY "kake" TO DECRYPT RECEIVED DATA
S410
Ra' = Ra ?
NO
YES
S411
MEDIA-2 ID' = MEDIA-2 ID ?
NO
YES
S412
GENERATE RANDOM NUMBERS "Kses"
3
S413
REGARD MUTUAL AUTHENTICATION AS FAILURE
END

FIG. 24

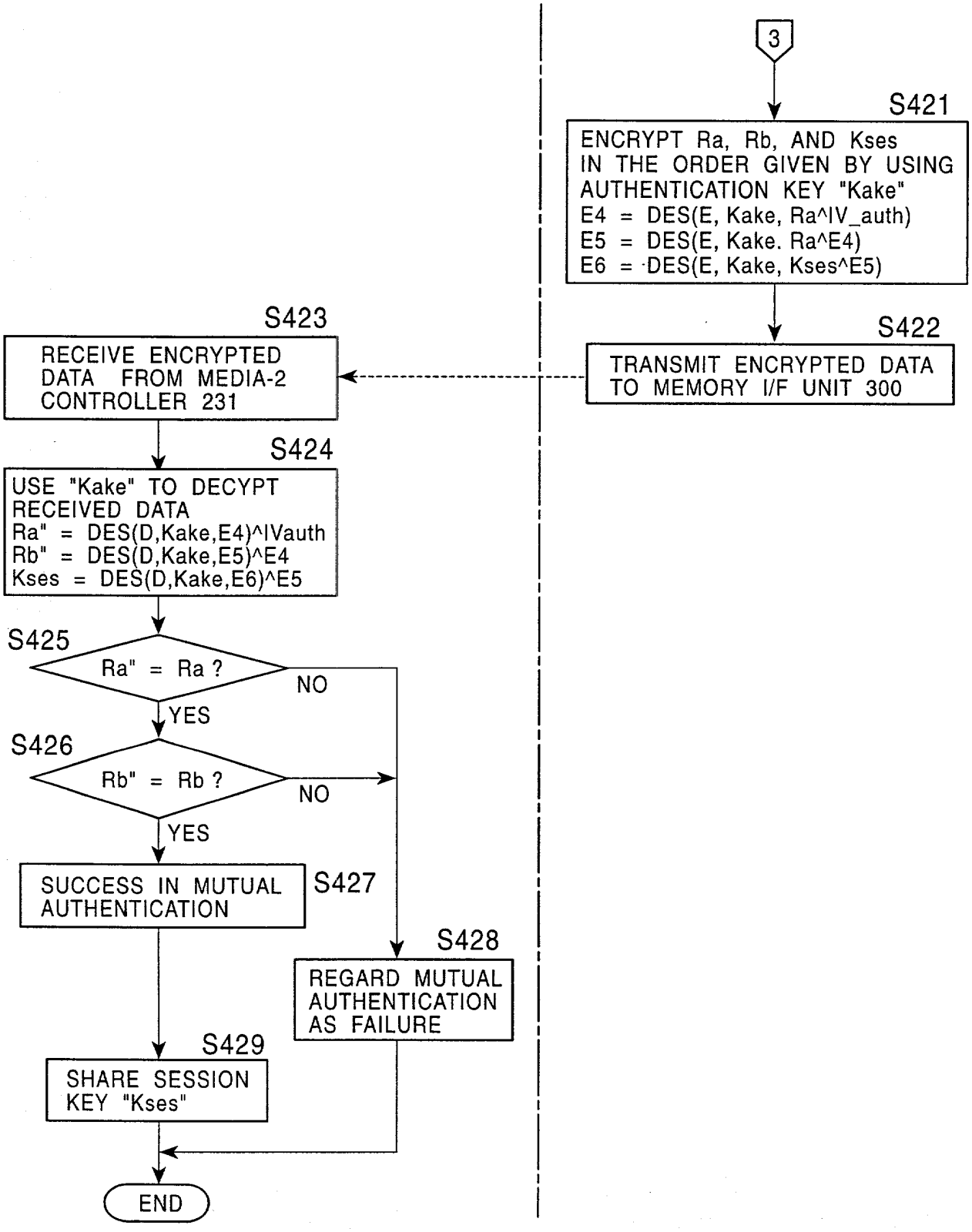

3
S421
ENCRYPT Ra, Rb, AND Kses
IN THE ORDER GIVEN BY USING
AUTHENTICATION KEY "Kake"
E4 = DES(E, Kake, Ra^IV_auth)
E5 = DES(E, Kake. Ra^E4)
E6 = DES(E, Kake, Kses^E5)
S422
TRANSMIT ENCRYPTED DATA
TO MEMORY I/F UNIT 300
S423
RECEIVE ENCRYPTED
DATA FROM MEDIA-2
CONTROLLER 231
S424
USE "Kake" TO DECYPT
RECEIVED DATA
Ra" = DES(D,Kake,E4)^IVauth
Rb" = DES(D,Kake,E5)^E4
Kses = DES(D,Kake,E6)^E5
S425
Ra" = Ra ?
NO
YES
S426
Rb" = Rb ?
NO
YES
SUCCESS IN MUTUAL
AUTHENTICATION
S427
S428
REGARD MUTUAL
AUTHENTICATION
AS FAILURE
S429
SHARE SESSION
KEY "Kses"
END

FIG. 25

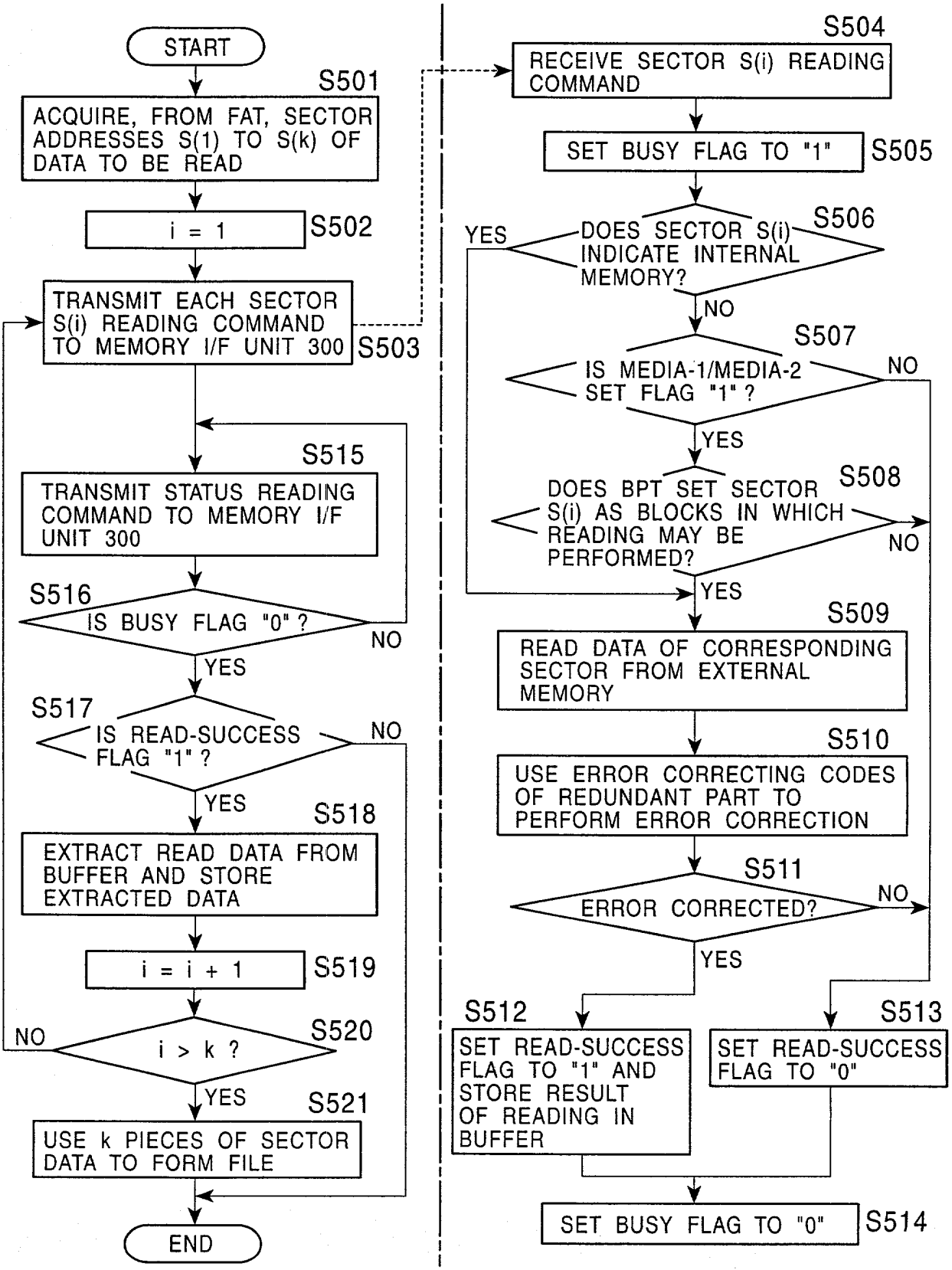

START
S501
ACQUIRE, FROM FAT, SECTOR ADDRESSES S(1) TO S(k) OF DATA TO BE READ
i = 1
S502
TRANSMIT EACH SECTOR S(i) READING COMMAND TO MEMORY I/F UNIT 300
S503
S515
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S516
IS BUSY FLAG "0" ?
NO
YES
S517
IS READ-SUCCESS FLAG "1" ?
NO
YES
S518
EXTRACT READ DATA FROM BUFFER AND STORE EXTRACTED DATA
i = i + 1
S519
NO
i > k ?
S520
YES
S521
USE k PIECES OF SECTOR DATA TO FORM FILE
END
S504
RECEIVE SECTOR S(i) READING COMMAND
SET BUSY FLAG TO "1"
S505
YES
DOES SECTOR S(i) INDICATE INTERNAL MEMORY?
S506
NO
S507
IS MEDIA-1/MEDIA-2 SET FLAG "1" ?
NO
YES
DOES BPT SET SECTOR S(i) AS BLOCKS IN WHICH READING MAY BE PERFORMED?
S508
NO
YES
S509
READ DATA OF CORRESPONDING SECTOR FROM EXTERNAL MEMORY
S510
USE ERROR CORRECTING CODES OF REDUNDANT PART TO PERFORM ERROR CORRECTION
S511
ERROR CORRECTED?
NO
YES
S512
SET READ-SUCCESS FLAG TO "1" AND STORE RESULT OF READING IN BUFFER
S513
SET READ-SUCCESS FLAG TO "0"
SET BUSY FLAG TO "0"
S514

FIG. 26

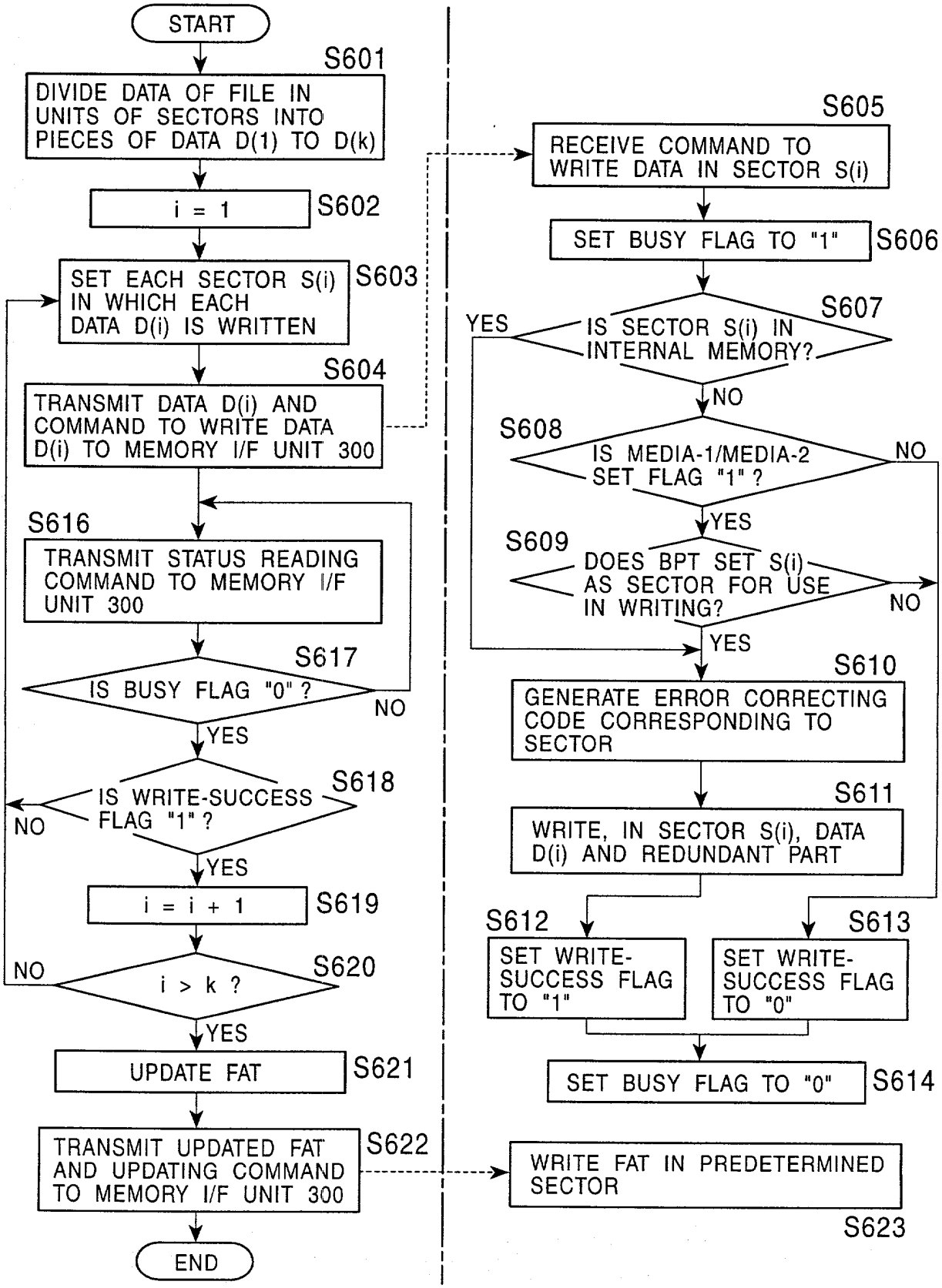

START
S601
DIVIDE DATA OF FILE IN UNITS OF SECTORS INTO PIECES OF DATA D(1) TO D(k)
i = 1
S602
SET EACH SECTOR S(i) IN WHICH EACH DATA D(i) IS WRITTEN
S603
TRANSMIT DATA D(i) AND COMMAND TO WRITE DATA D(i) TO MEMORY I/F UNIT 300
S604
S616
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S617
IS BUSY FLAG "0" ?
NO
YES
S618
IS WRITE-SUCCESS FLAG "1" ?
NO
YES
i = i + 1
S619
NO
i > k ?
S620
YES
UPDATE FAT
S621
TRANSMIT UPDATED FAT AND UPDATING COMMAND TO MEMORY I/F UNIT 300
S622
END
S605
RECEIVE COMMAND TO WRITE DATA IN SECTOR S(i)
SET BUSY FLAG TO "1"
S606
S607
YES
IS SECTOR S(i) IN INTERNAL MEMORY?
NO
S608
IS MEDIA-1/MEDIA-2 SET FLAG "1" ?
NO
YES
S609
DOES BPT SET S(i) AS SECTOR FOR USE IN WRITING?
NO
YES
S610
GENERATE ERROR CORRECTING CODE CORRESPONDING TO SECTOR
S611
WRITE, IN SECTOR S(i), DATA D(i) AND REDUNDANT PART
S612
SET WRITE-SUCCESS FLAG TO "1"
S613
SET WRITE-SUCCESS FLAG TO "0"
SET BUSY FLAG TO "0"
S614
WRITE FAT IN PREDETERMINED SECTOR
S623

FIG. 27A

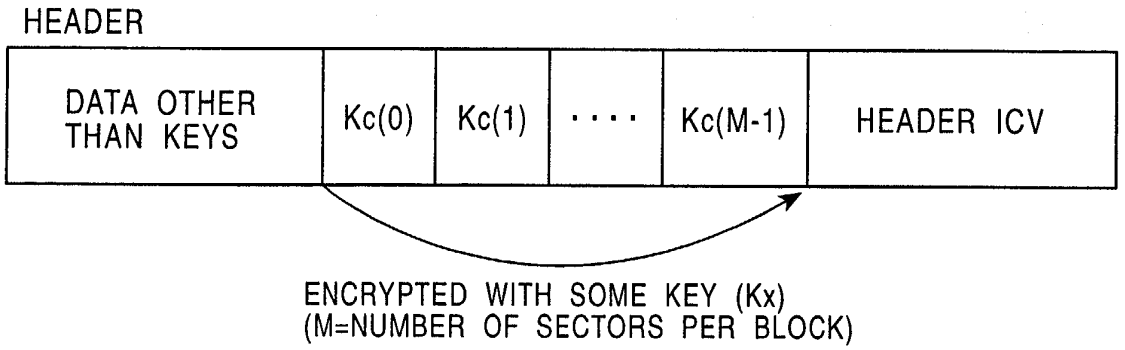


FIG. 27B

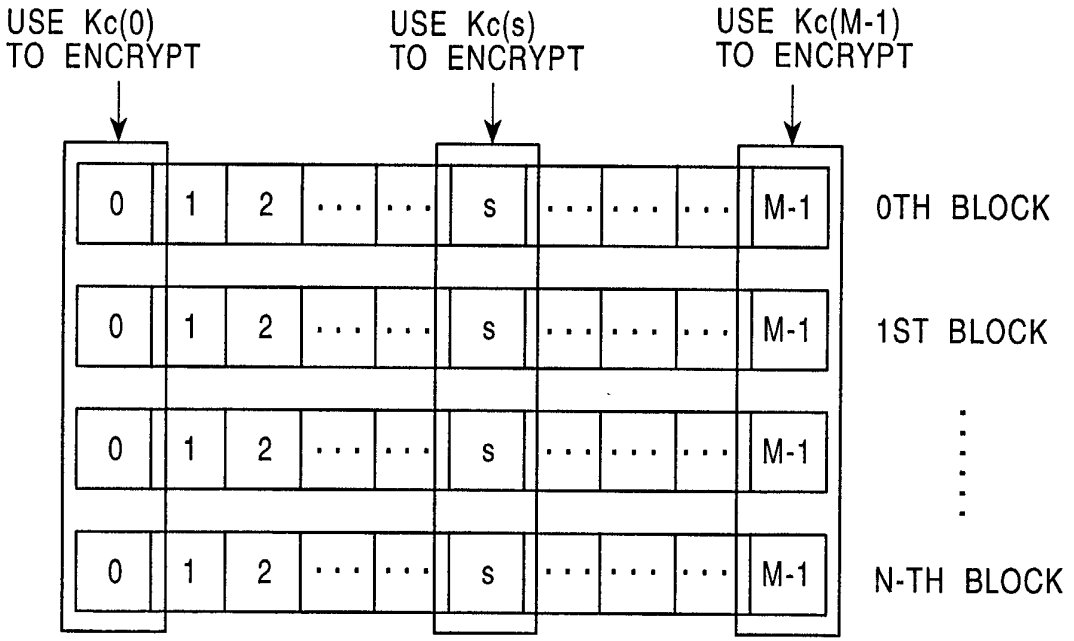


FIG. 28A

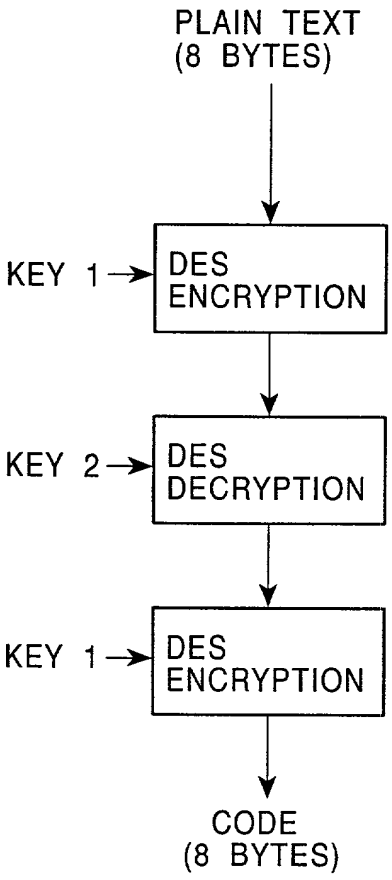


FIG. 28B

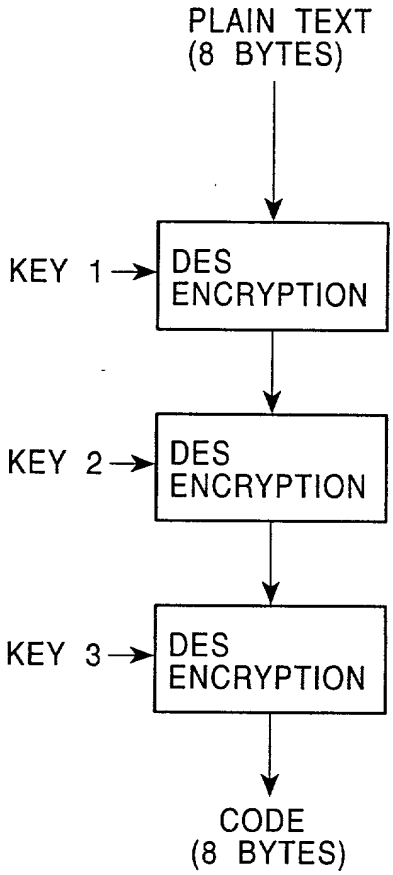


FIG. 29

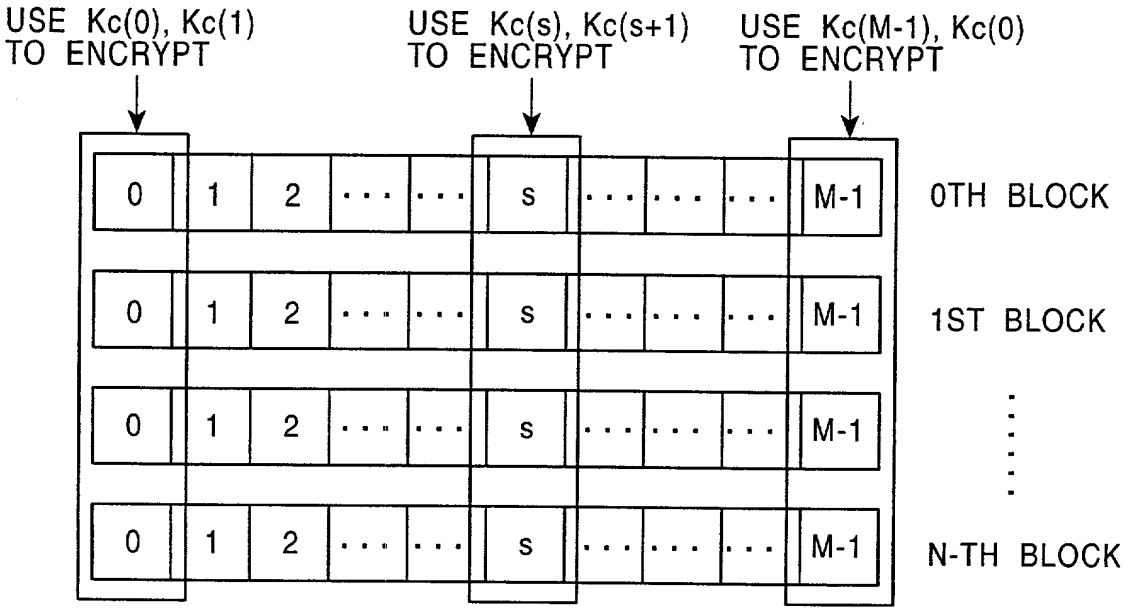

USE Kc(0), Kc(1) TO ENCRYPT
USE Kc(s), Kc(s+1) TO ENCRYPT
USE Kc(M-1), Kc(0) TO ENCRYPT
0 1 2 ... ... s M-1
0TH BLOCK
1ST BLOCK
N-TH BLOCK

FIG. 30

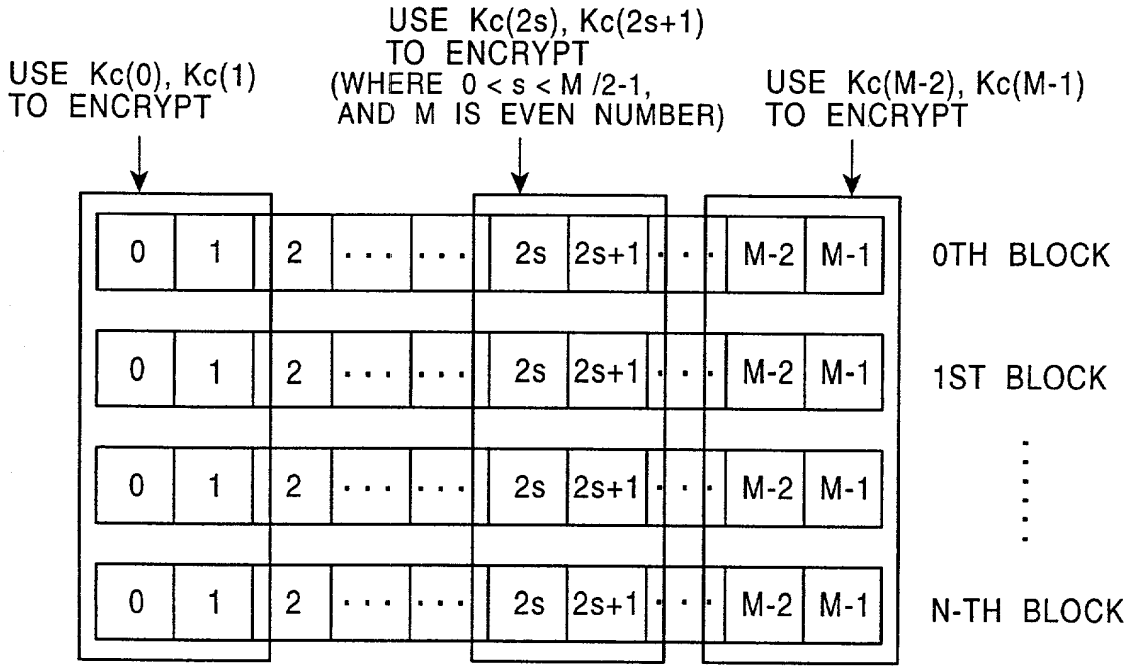

USE Kc(0), Kc(1)
TO ENCRYPT
USE Kc(2s), Kc(2s+1)
TO ENCRYPT
(WHERE 0 < s < M /2-1,
AND M IS EVEN NUMBER)
USE Kc(M-2), Kc(M-1)
TO ENCRYPT
0 1 2 · · · · · · 2s 2s+1 · · · M-2 M-1
0TH BLOCK
1ST BLOCK
N-TH BLOCK

FIG. 31

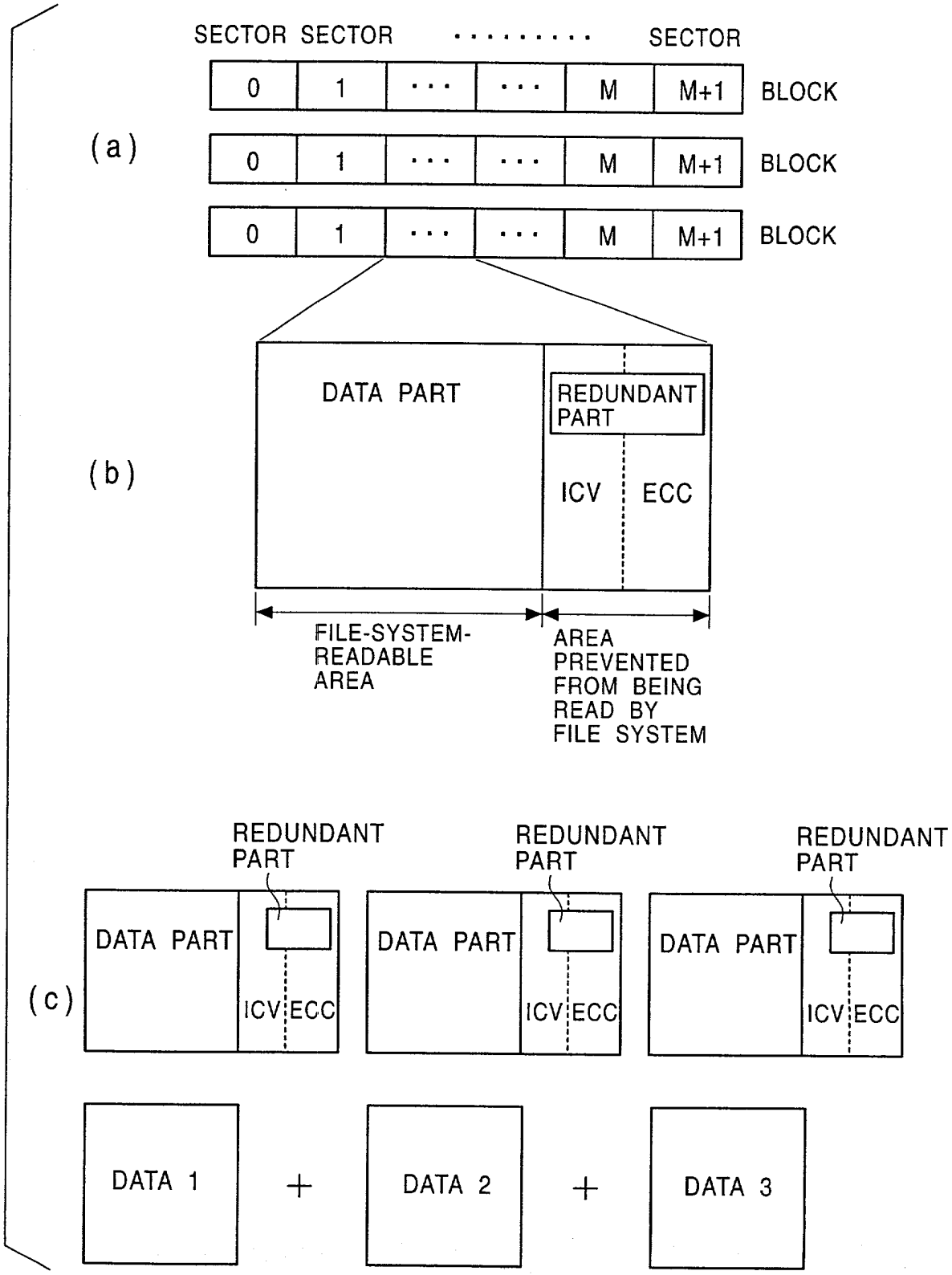

SECTOR SECTOR ..... SECTOR
0
1
M
M+1
BLOCK
(a)
DATA PART
REDUNDANT PART
ICV
ECC
(b)
FILE-SYSTEM-READABLE AREA
AREA PREVENTED FROM BEING READ BY FILE SYSTEM
(c)
DATA 1
DATA 2
DATA 3

FIG. 32

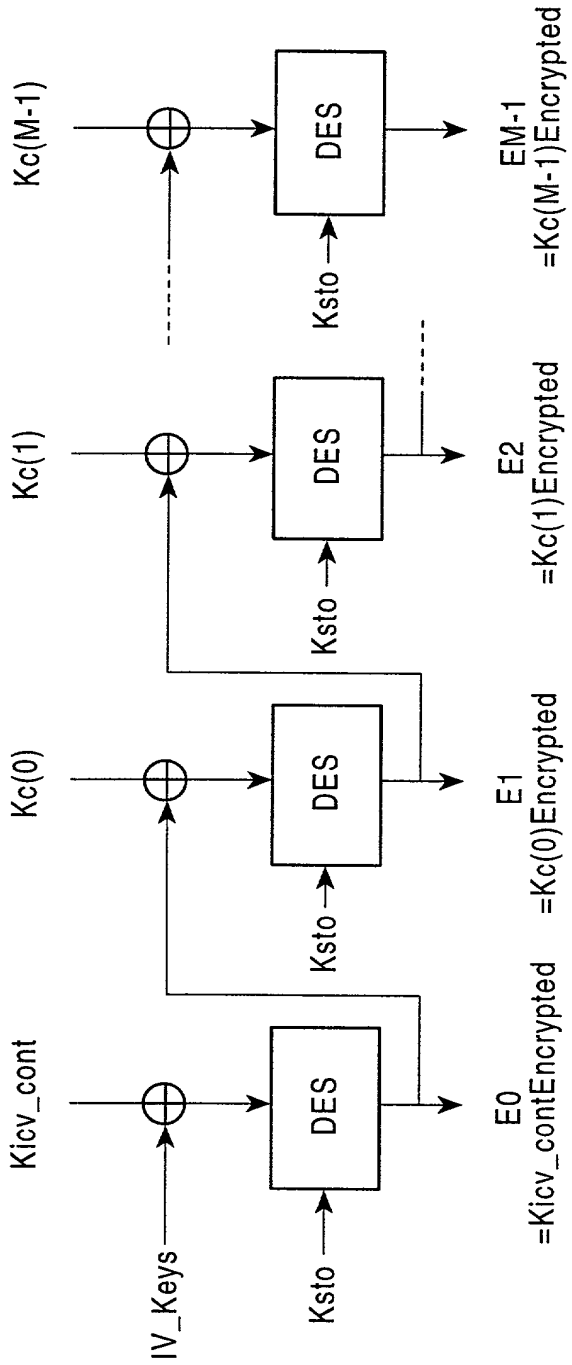

Kicv_cont
Kc(0)
Kc(1)
Kc(M-1)
IV_Keys
Ksto
DES
E0
=Kicv_contEncrypted
E1
=Kc(0)Encrypted
E2
=Kc(1)Encrypted
EM-1
=Kc(M-1)Encrypted

FIG. 33

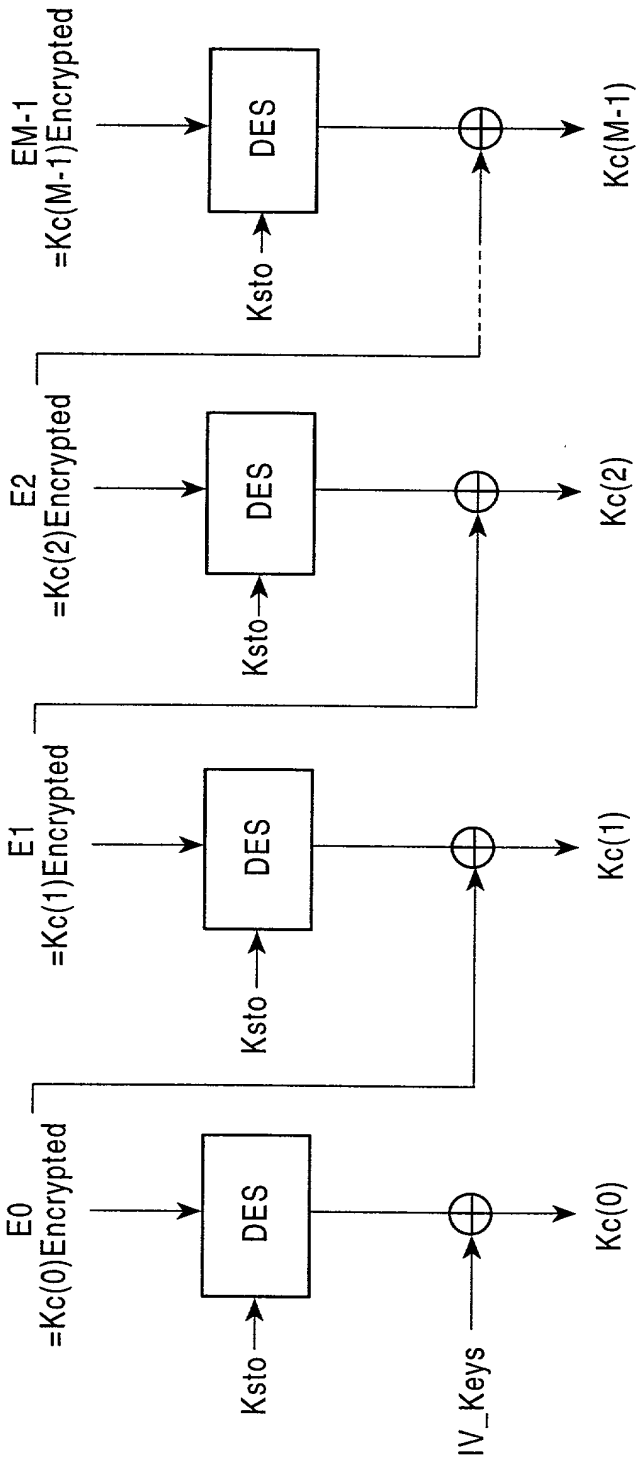

E0
=Kc(0)Encrypted
DES
Ksto
IV_Keys
Kc(0)
E1
=Kc(1)Encrypted
DES
Ksto
Kc(1)
E2
=Kc(2)Encrypted
DES
Ksto
Kc(2)
EM-1
=Kc(M-1)Encrypted
DES
Ksto
Kc(M-1)

FIG. 34

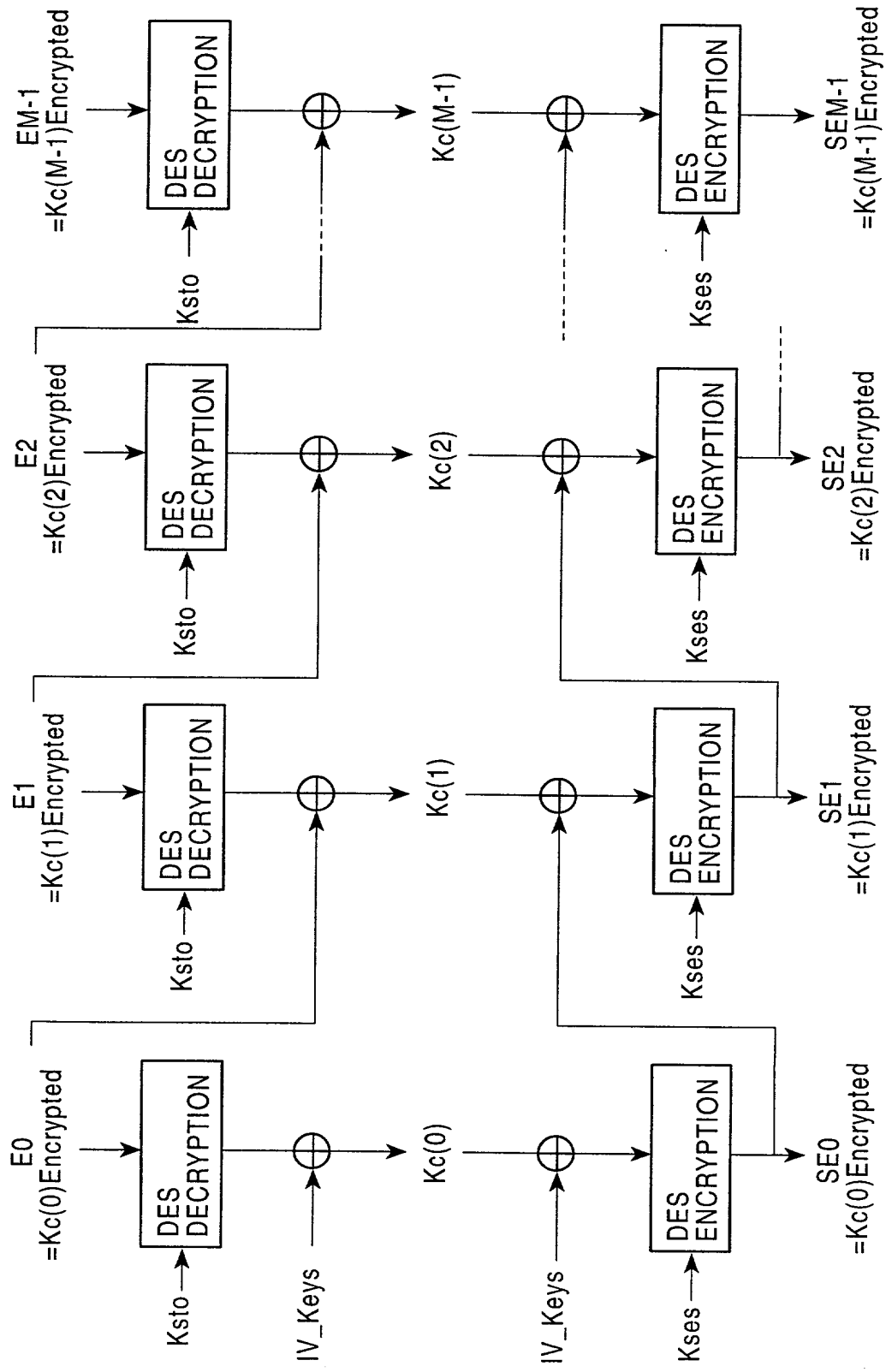

E0
=Kc(0)Encrypted
DES
DECRYPTION
Ksto
IV_Keys
Kc(0)
DES
ENCRYPTION
Kses
SE0
=Kc(0)Encrypted
E1
=Kc(1)Encrypted
DES
DECRYPTION
Ksto
Kc(1)
DES
ENCRYPTION
Kses
SE1
=Kc(1)Encrypted
E2
=Kc(2)Encrypted
DES
DECRYPTION
Ksto
Kc(2)
DES
ENCRYPTION
Kses
SE2
=Kc(2)Encrypted
EM-1
=Kc(M-1)Encrypted
DES
DECRYPTION
Ksto
Kc(M-1)
DES
ENCRYPTION
Kses
SEM-1
=Kc(M-1)Encrypted

FIG. 35

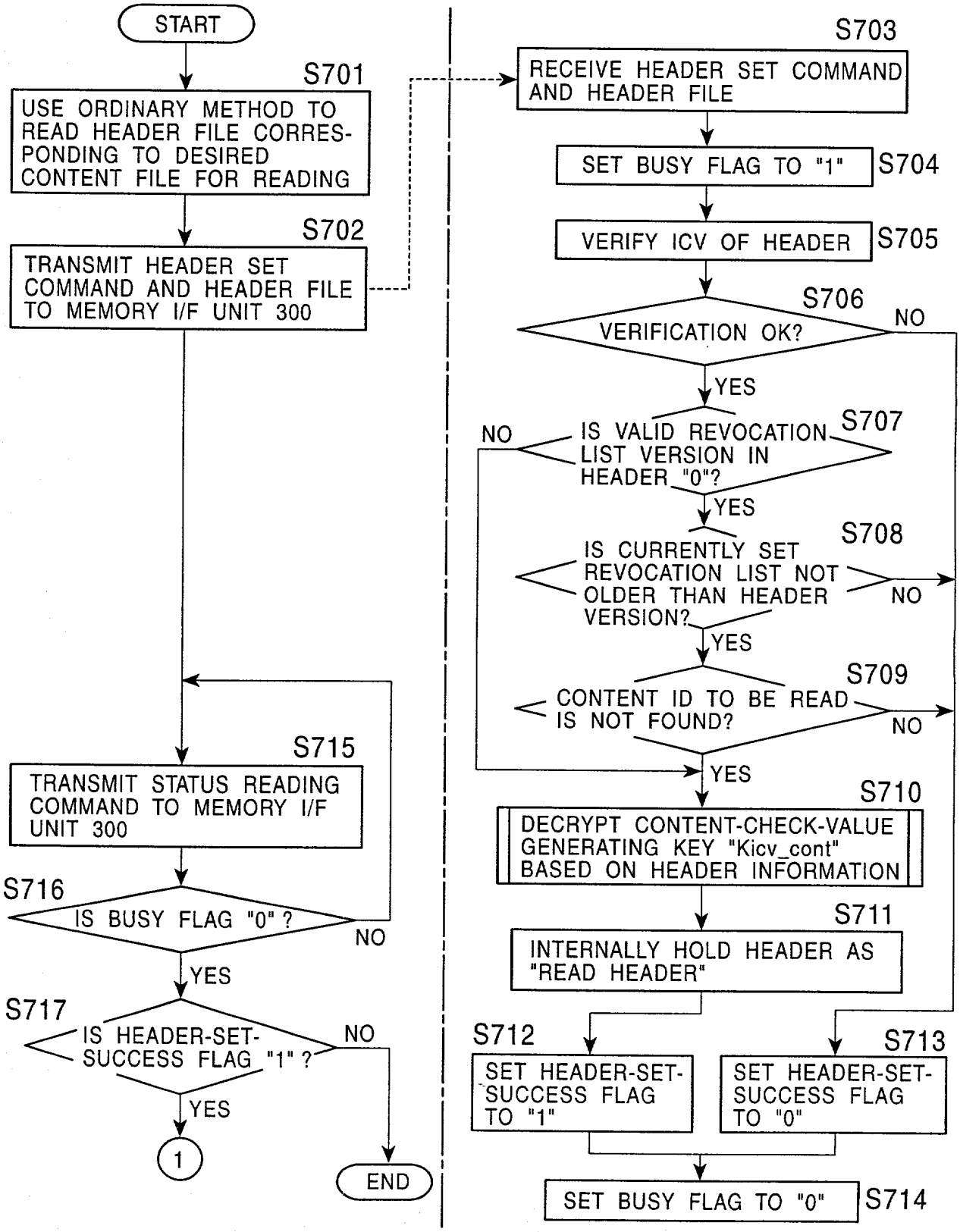

START
S701
USE ORDINARY METHOD TO READ HEADER FILE CORRESPONDING TO DESIRED CONTENT FILE FOR READING
S702
TRANSMIT HEADER SET COMMAND AND HEADER FILE TO MEMORY I/F UNIT 300
S715
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S716
IS BUSY FLAG "0" ?
NO
YES
S717
IS HEADER-SET-SUCCESS FLAG "1" ?
NO
YES
1
END
S703
RECEIVE HEADER SET COMMAND AND HEADER FILE
SET BUSY FLAG TO "1"
S704
VERIFY ICV OF HEADER
S705
S706
VERIFICATION OK?
NO
YES
S707
IS VALID REVOCATION LIST VERSION IN HEADER "0"?
NO
YES
S708
IS CURRENTLY SET REVOCATION LIST NOT OLDER THAN HEADER VERSION?
NO
YES
S709
CONTENT ID TO BE READ IS NOT FOUND?
NO
YES
S710
DECRYPT CONTENT-CHECK-VALUE GENERATING KEY "Kicv_cont" BASED ON HEADER INFORMATION
S711
INTERNALLY HOLD HEADER AS "READ HEADER"
S712
SET HEADER-SET-SUCCESS FLAG TO "1"
S713
SET HEADER-SET-SUCCESS FLAG TO "0"
SET BUSY FLAG TO "0"
S714

FIG. 36

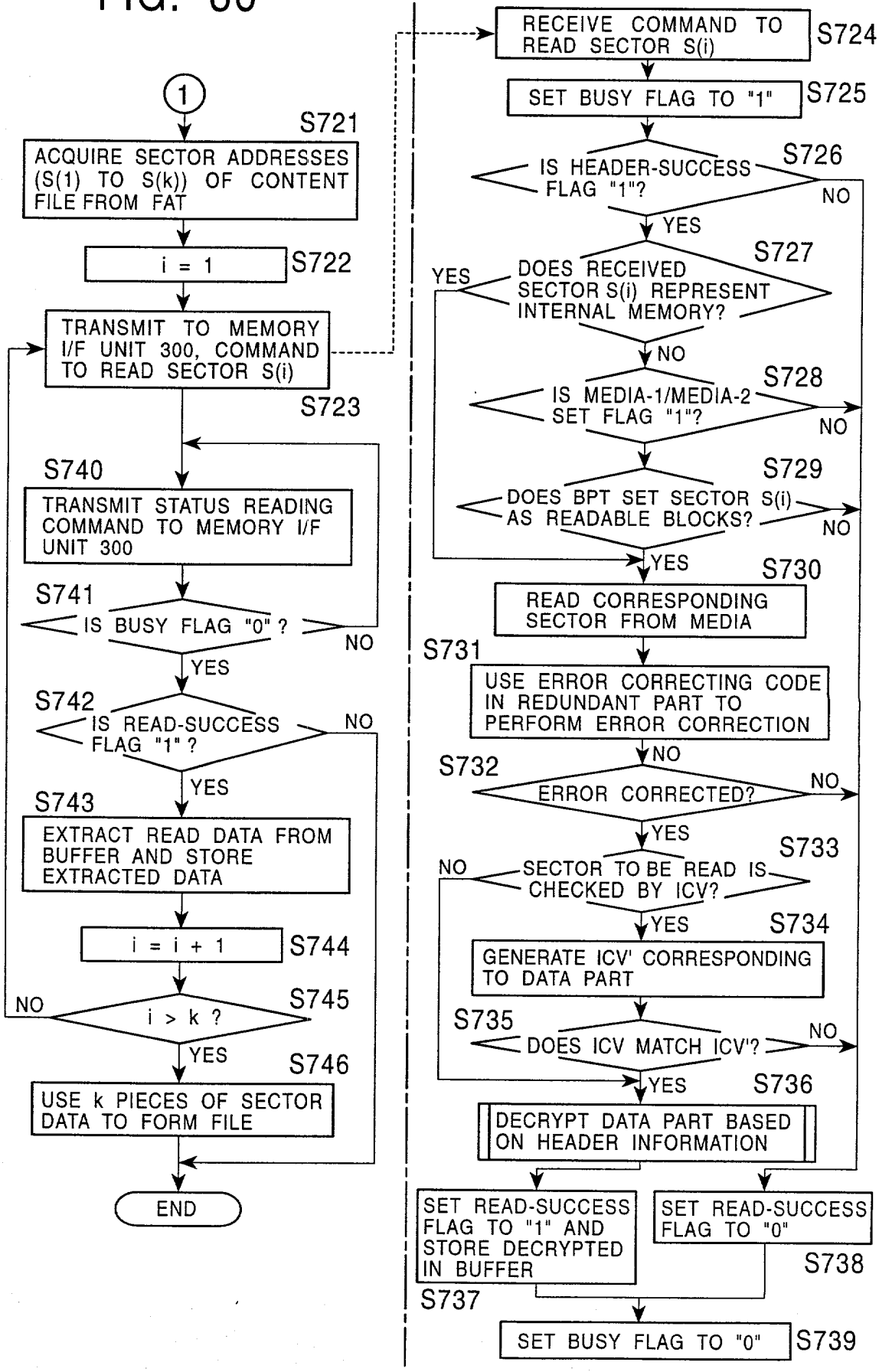

1
S721
ACQUIRE SECTOR ADDRESSES (S(1) TO S(k)) OF CONTENT FILE FROM FAT
i = 1
S722
TRANSMIT TO MEMORY I/F UNIT 300, COMMAND TO READ SECTOR S(i)
S723
S740
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S741
IS BUSY FLAG "0" ?
NO
YES
S742
IS READ-SUCCESS FLAG "1" ?
NO
YES
S743
EXTRACT READ DATA FROM BUFFER AND STORE EXTRACTED DATA
i = i + 1
S744
NO
i > k ?
S745
YES
S746
USE k PIECES OF SECTOR DATA TO FORM FILE
END
RECEIVE COMMAND TO READ SECTOR S(i)
S724
SET BUSY FLAG TO "1"
S725
S726
IS HEADER-SUCCESS FLAG "1"?
NO
YES
S727
YES
DOES RECEIVED SECTOR S(i) REPRESENT INTERNAL MEMORY?
NO
S728
IS MEDIA-1/MEDIA-2 SET FLAG "1"?
NO
S729
DOES BPT SET SECTOR S(i) AS READABLE BLOCKS?
NO
YES
S730
READ CORRESPONDING SECTOR FROM MEDIA
S731
USE ERROR CORRECTING CODE IN REDUNDANT PART TO PERFORM ERROR CORRECTION
NO
S732
ERROR CORRECTED?
NO
YES
S733
NO
SECTOR TO BE READ IS CHECKED BY ICV?
YES
S734
GENERATE ICV' CORRESPONDING TO DATA PART
S735
DOES ICV MATCH ICV'?
NO
YES
S736
DECRYPT DATA PART BASED ON HEADER INFORMATION
SET READ-SUCCESS FLAG TO "1" AND STORE DECRYPTED IN BUFFER
S737
SET READ-SUCCESS FLAG TO "0"
S738
SET BUSY FLAG TO "0"
S739

FIG. 37

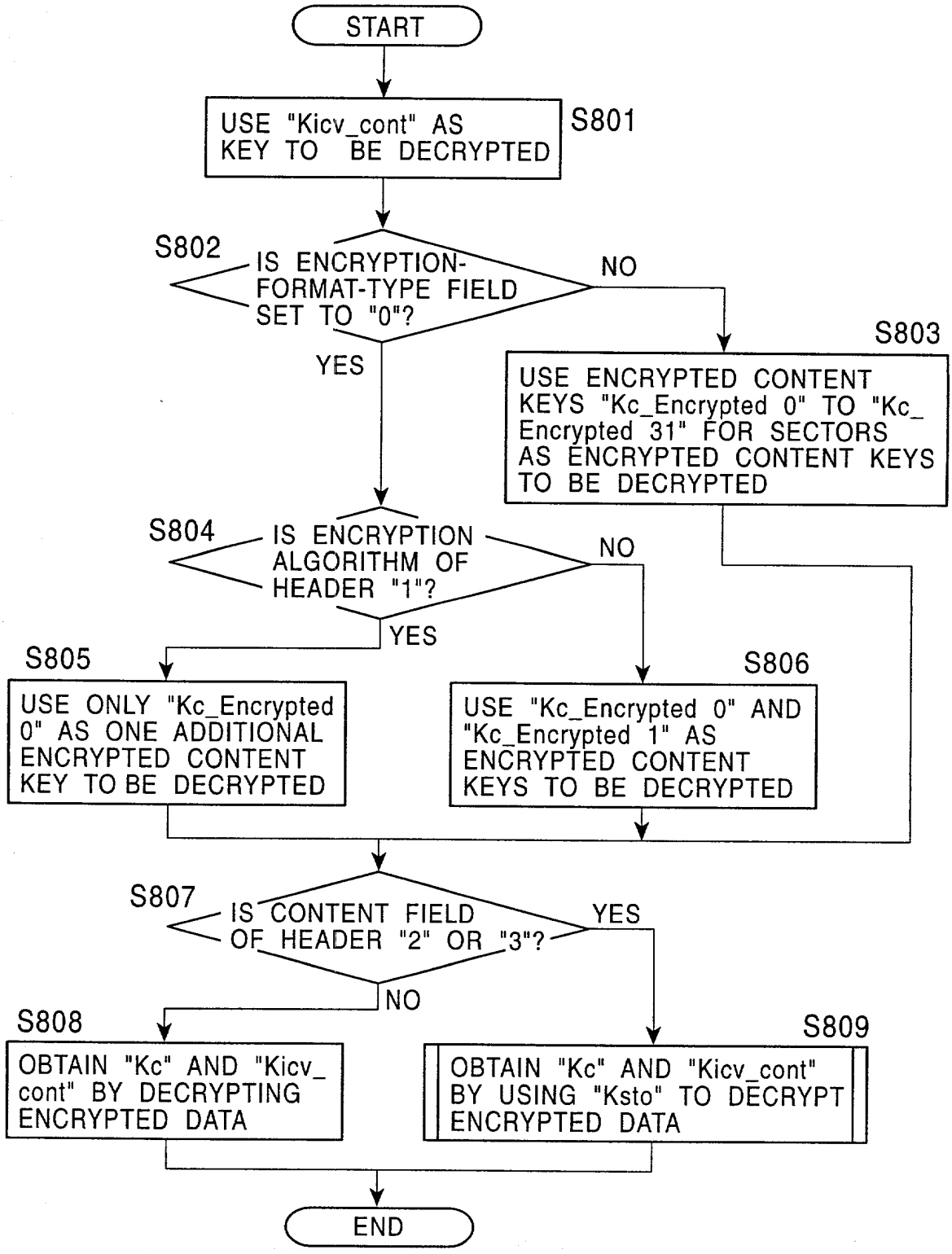

START
USE "Kicv_cont" AS KEY TO BE DECRYPTED
S801
S802
IS ENCRYPTION-FORMAT-TYPE FIELD SET TO "0"?
NO
YES
S803
USE ENCRYPTED CONTENT KEYS "Kc_Encrypted 0" TO "Kc_Encrypted 31" FOR SECTORS AS ENCRYPTED CONTENT KEYS TO BE DECRYPTED
S804
IS ENCRYPTION ALGORITHM OF HEADER "1"?
NO
YES
S805
USE ONLY "Kc_Encrypted 0" AS ONE ADDITIONAL ENCRYPTED CONTENT KEY TO BE DECRYPTED
S806
USE "Kc_Encrypted 0" AND "Kc_Encrypted 1" AS ENCRYPTED CONTENT KEYS TO BE DECRYPTED
S807
IS CONTENT FIELD OF HEADER "2" OR "3"?
YES
NO
S808
OBTAIN "Kc" AND "Kicv_cont" BY DECRYPTING ENCRYPTED DATA
S809
OBTAIN "Kc" AND "Kicv_cont" BY USING "Ksto" TO DECRYPT ENCRYPTED DATA
END

FIG. 38

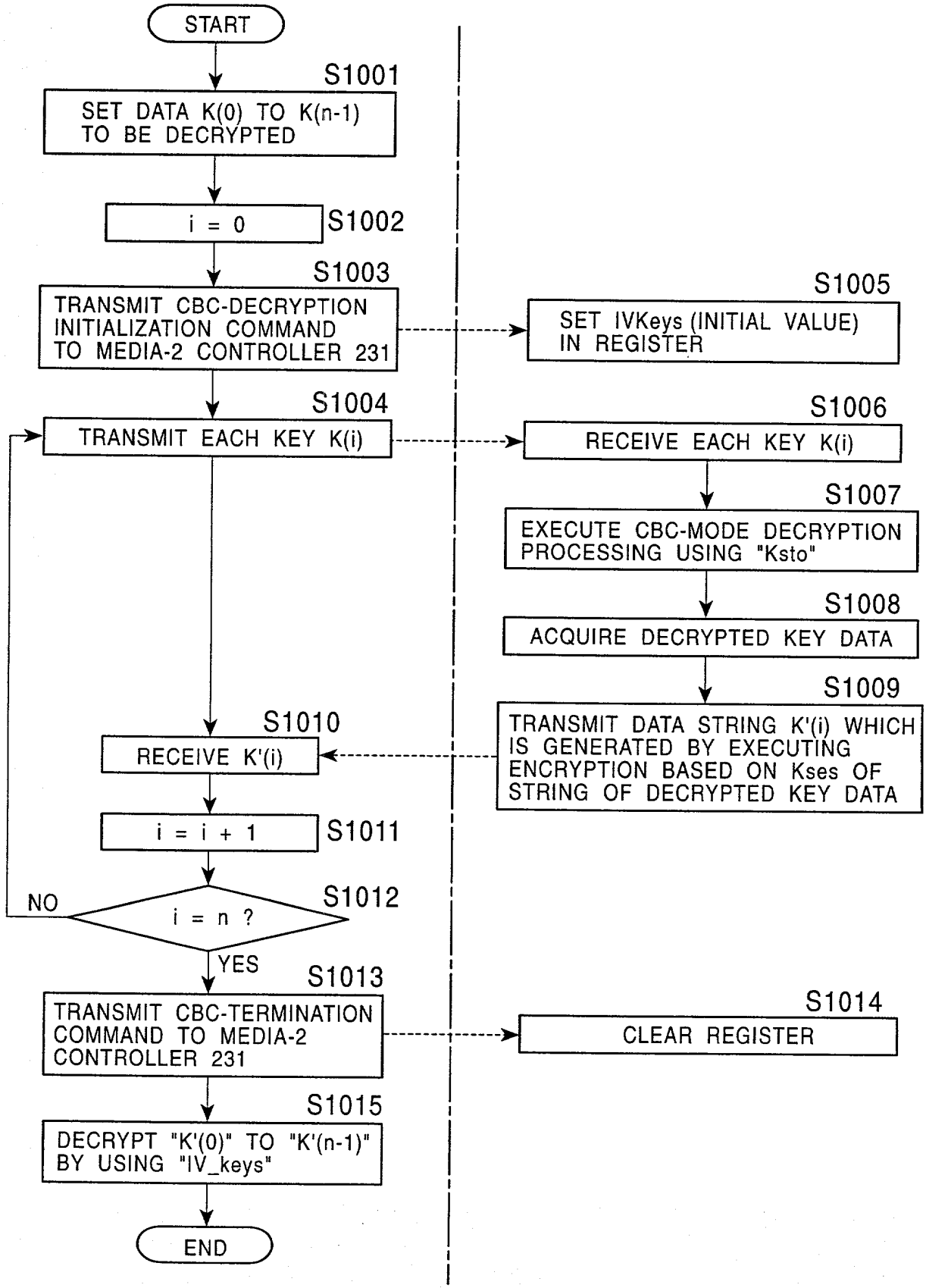

START
S1001
SET DATA K(0) TO K(n-1) TO BE DECRYPTED
i = 0
S1002
S1003
TRANSMIT CBC-DECRYPTION INITIALIZATION COMMAND TO MEDIA-2 CONTROLLER 231
S1005
SET IVKeys (INITIAL VALUE) IN REGISTER
S1004
TRANSMIT EACH KEY K(i)
S1006
RECEIVE EACH KEY K(i)
S1007
EXECUTE CBC-MODE DECRYPTION PROCESSING USING "Ksto"
S1008
ACQUIRE DECRYPTED KEY DATA
S1009
TRANSMIT DATA STRING K'(i) WHICH IS GENERATED BY EXECUTING ENCRYPTION BASED ON Kses OF STRING OF DECRYPTED KEY DATA
S1010
RECEIVE K'(i)
i = i + 1
S1011
NO
i = n ?
S1012
YES
S1013
TRANSMIT CBC-TERMINATION COMMAND TO MEDIA-2 CONTROLLER 231
S1014
CLEAR REGISTER
S1015
DECRYPT "K'(0)" TO "K'(n-1)" BY USING "IV_keys"
END

FIG. 39

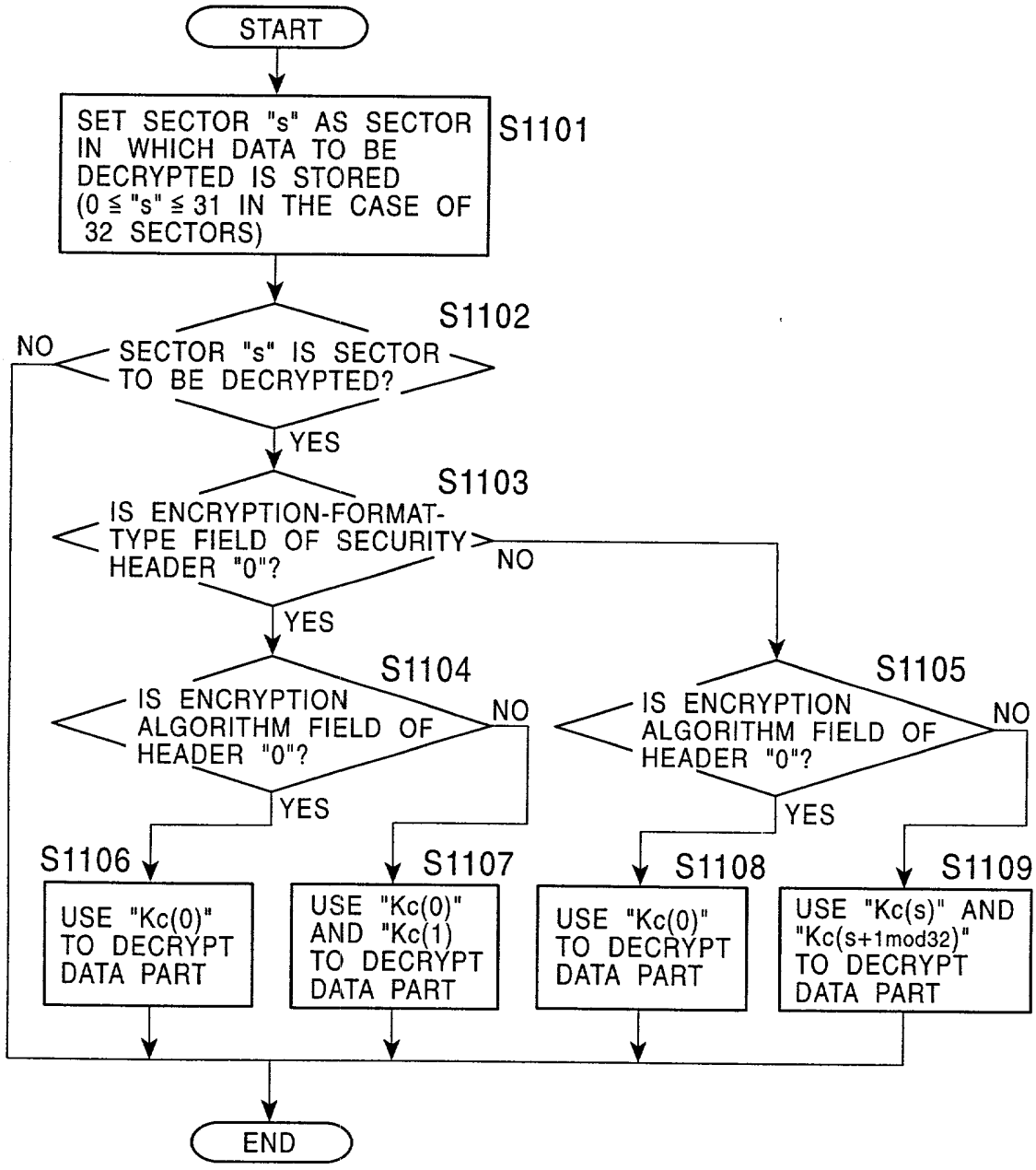

START
SET SECTOR "s" AS SECTOR IN WHICH DATA TO BE DECRYPTED IS STORED (0 ≦ "s" ≦ 31 IN THE CASE OF 32 SECTORS)
S1101
S1102
SECTOR "s" IS SECTOR TO BE DECRYPTED?
NO
YES
S1103
IS ENCRYPTION-FORMAT-TYPE FIELD OF SECURITY HEADER "0"?
NO
YES
S1104
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1105
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1106
USE "Kc(0)" TO DECRYPT DATA PART
S1107
USE "Kc(0)" AND "Kc(1) TO DECRYPT DATA PART
S1108
USE "Kc(0)" TO DECRYPT DATA PART
S1109
USE "Kc(s)" AND "Kc(s+1mod32)" TO DECRYPT DATA PART
END

FIG. 40

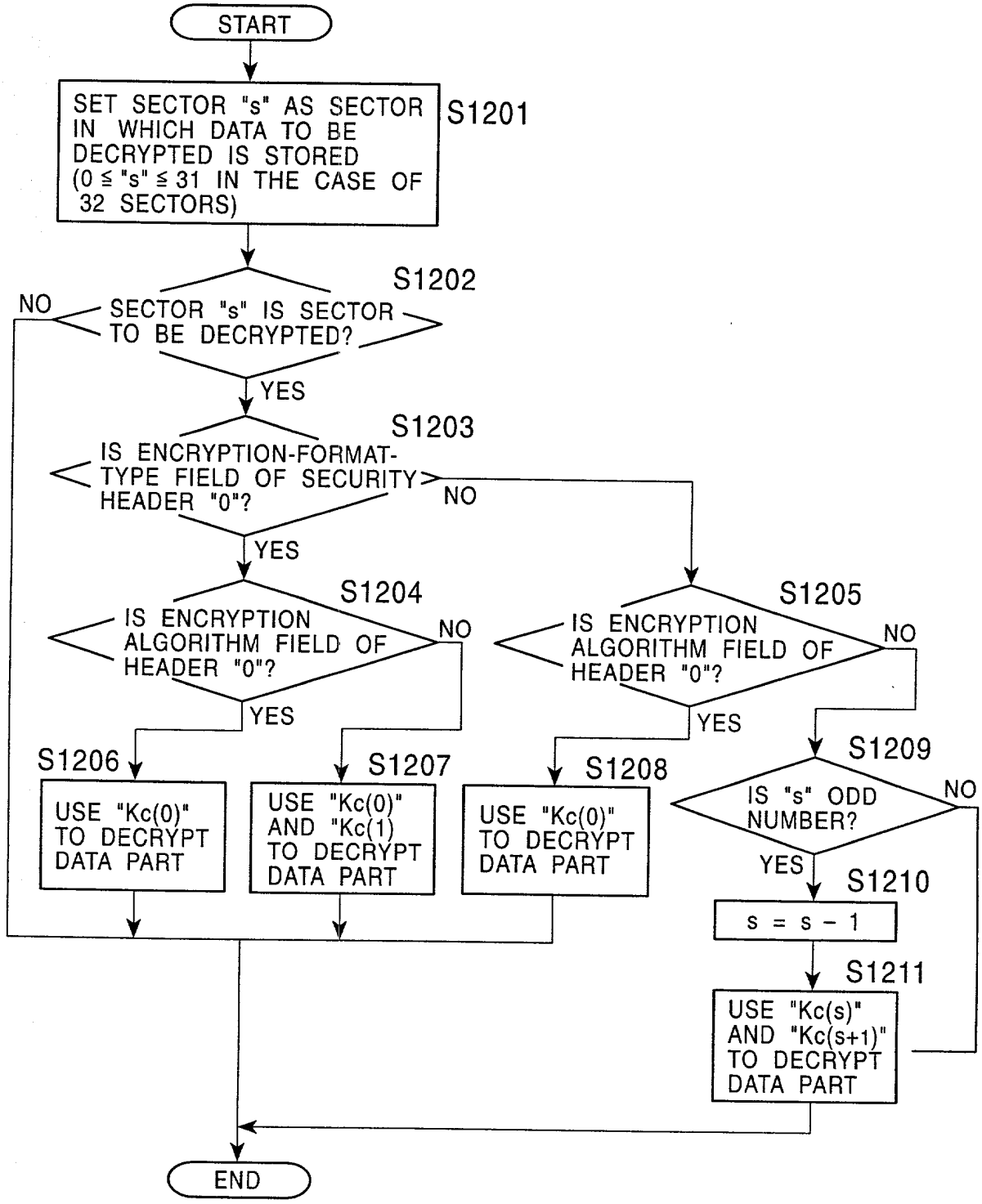

START
SET SECTOR "s" AS SECTOR IN WHICH DATA TO BE DECRYPTED IS STORED (0 ≦ "s" ≦ 31 IN THE CASE OF 32 SECTORS)
S1201
S1202
SECTOR "s" IS SECTOR TO BE DECRYPTED?
NO
YES
S1203
IS ENCRYPTION-FORMAT-TYPE FIELD OF SECURITY HEADER "0"?
NO
YES
S1204
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1205
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1206
USE "Kc(0)" TO DECRYPT DATA PART
S1207
USE "Kc(0)" AND "Kc(1) TO DECRYPT DATA PART
S1208
USE "Kc(0)" TO DECRYPT DATA PART
S1209
IS "s" ODD NUMBER?
NO
YES
S1210
s = s - 1
S1211
USE "Kc(s)" AND "Kc(s+1)" TO DECRYPT DATA PART
END

FIG. 41

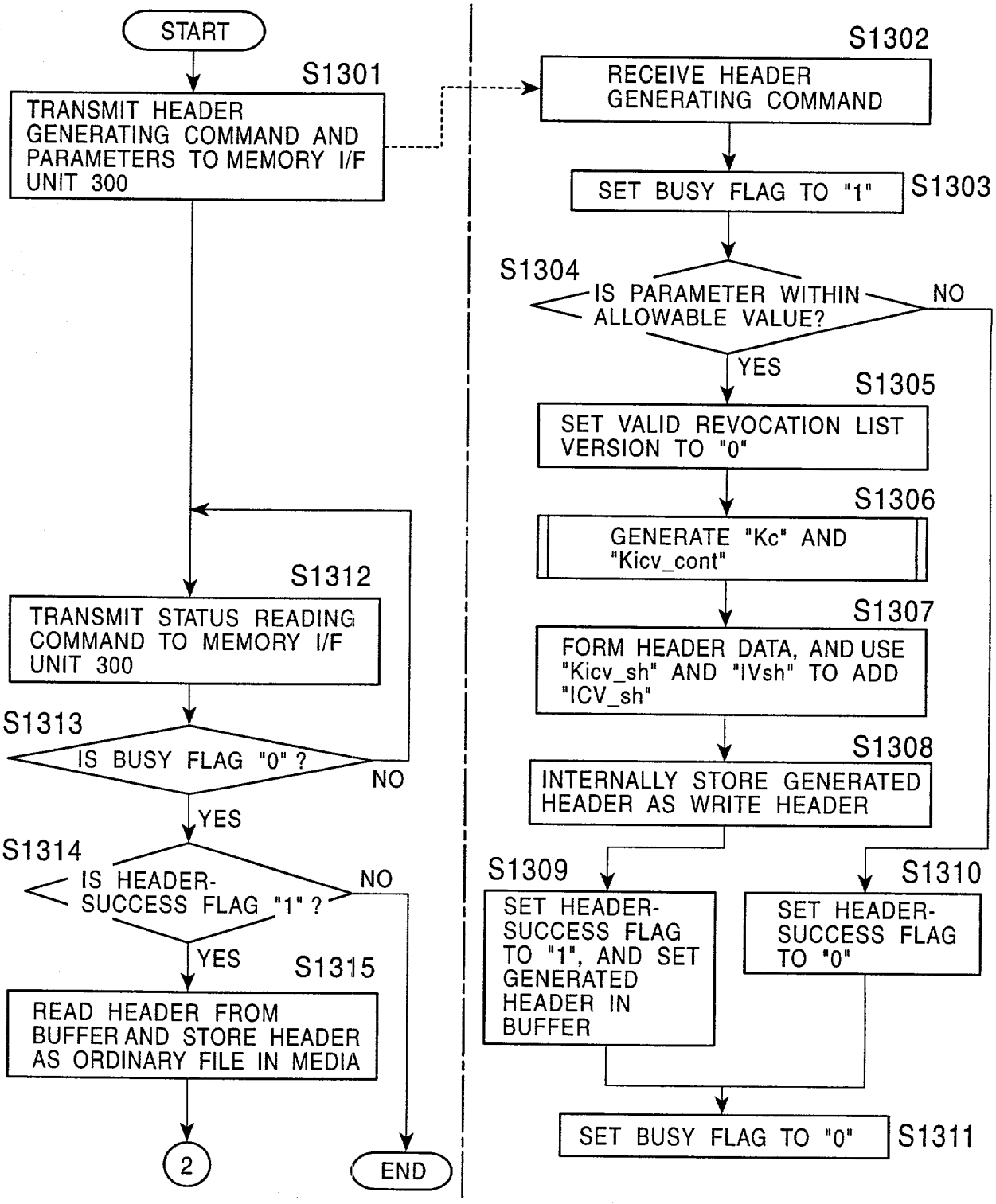

START
S1301
TRANSMIT HEADER GENERATING COMMAND AND PARAMETERS TO MEMORY I/F UNIT 300
S1312
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S1313
IS BUSY FLAG "0" ?
NO
YES
S1314
IS HEADER-SUCCESS FLAG "1" ?
NO
YES
S1315
READ HEADER FROM BUFFER AND STORE HEADER AS ORDINARY FILE IN MEDIA
2
END
S1302
RECEIVE HEADER GENERATING COMMAND
SET BUSY FLAG TO "1"
S1303
S1304
IS PARAMETER WITHIN ALLOWABLE VALUE?
NO
YES
S1305
SET VALID REVOCATION LIST VERSION TO "0"
S1306
GENERATE "Kc" AND "Kicv_cont"
S1307
FORM HEADER DATA, AND USE "Kicv_sh" AND "IVsh" TO ADD "ICV_sh"
S1308
INTERNALLY STORE GENERATED HEADER AS WRITE HEADER
S1309
SET HEADER-SUCCESS FLAG TO "1", AND SET GENERATED HEADER IN BUFFER
S1310
SET HEADER-SUCCESS FLAG TO "0"
SET BUSY FLAG TO "0"
S1311

FIG. 42

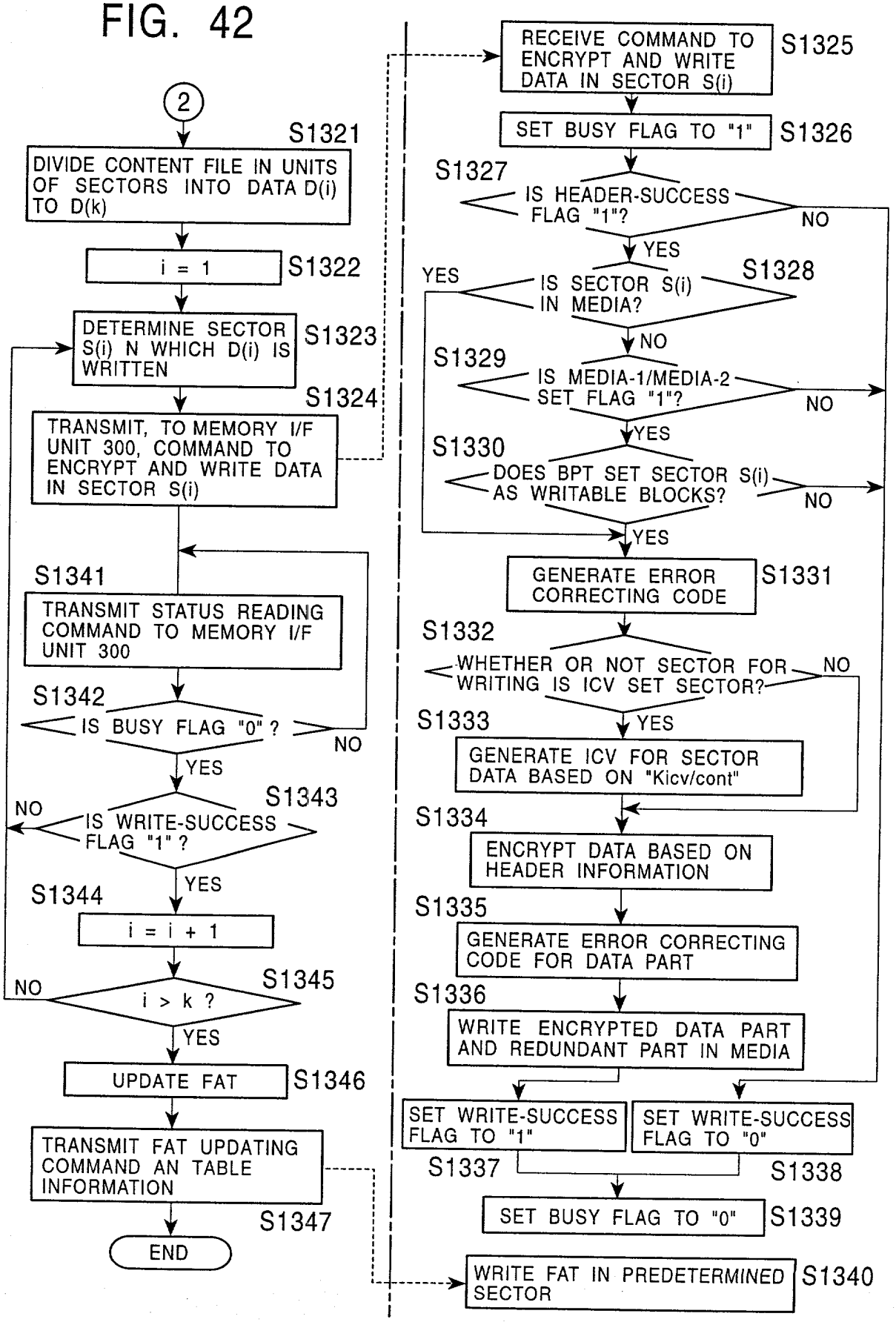

2
S1321
DIVIDE CONTENT FILE IN UNITS OF SECTORS INTO DATA D(i) TO D(k)
i = 1
S1322
DETERMINE SECTOR S(i) N WHICH D(i) IS WRITTEN
S1323
TRANSMIT, TO MEMORY I/F UNIT 300, COMMAND TO ENCRYPT AND WRITE DATA IN SECTOR S(i)
S1324
S1341
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S1342
IS BUSY FLAG "0" ?
NO
YES
S1343
IS WRITE-SUCCESS FLAG "1" ?
NO
YES
S1344
i = i + 1
S1345
i > k ?
NO
YES
UPDATE FAT
S1346
TRANSMIT FAT UPDATING COMMAND AN TABLE INFORMATION
S1347
END
RECEIVE COMMAND TO ENCRYPT AND WRITE DATA IN SECTOR S(i)
S1325
SET BUSY FLAG TO "1"
S1326
S1327
IS HEADER-SUCCESS FLAG "1"?
NO
YES
S1328
IS SECTOR S(i) IN MEDIA?
YES
NO
S1329
IS MEDIA-1/MEDIA-2 SET FLAG "1"?
NO
YES
S1330
DOES BPT SET SECTOR S(i) AS WRITABLE BLOCKS?
NO
YES
GENERATE ERROR CORRECTING CODE
S1331
S1332
WHETHER OR NOT SECTOR FOR WRITING IS ICV SET SECTOR?
NO
YES
S1333
GENERATE ICV FOR SECTOR DATA BASED ON "Kicv/cont"
S1334
ENCRYPT DATA BASED ON HEADER INFORMATION
S1335
GENERATE ERROR CORRECTING CODE FOR DATA PART
S1336
WRITE ENCRYPTED DATA PART AND REDUNDANT PART IN MEDIA
SET WRITE-SUCCESS FLAG TO "1"
S1337
SET WRITE-SUCCESS FLAG TO "0"
S1338
SET BUSY FLAG TO "0"
S1339
WRITE FAT IN PREDETERMINED SECTOR
S1340

FIG. 43

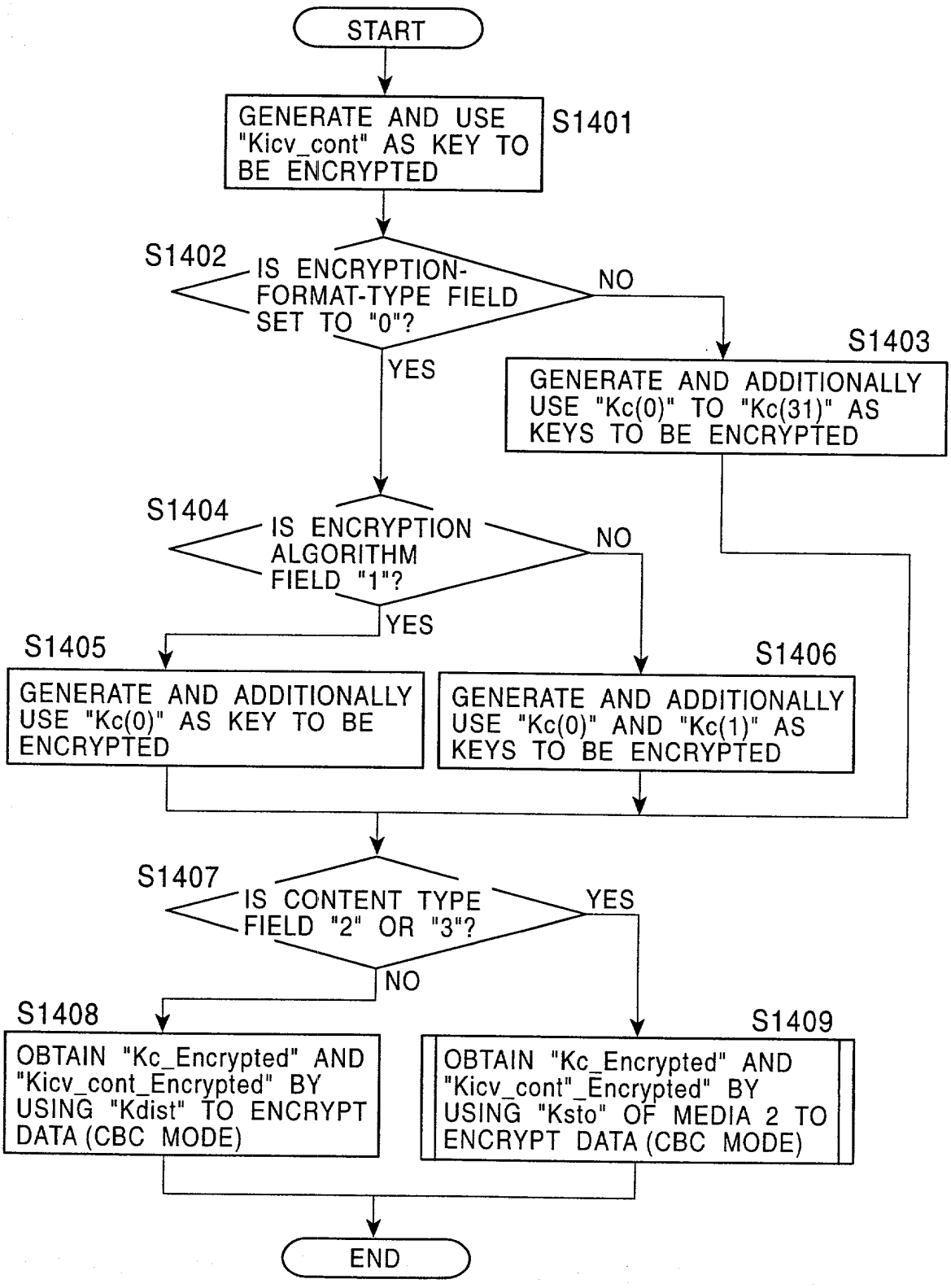

START
GENERATE AND USE "Kicv_cont" AS KEY TO BE ENCRYPTED
S1401
S1402
IS ENCRYPTION-FORMAT-TYPE FIELD SET TO "0"?
NO
YES
S1403
GENERATE AND ADDITIONALLY USE "Kc(0)" TO "Kc(31)" AS KEYS TO BE ENCRYPTED
S1404
IS ENCRYPTION ALGORITHM FIELD "1"?
NO
YES
S1405
GENERATE AND ADDITIONALLY USE "Kc(0)" AS KEY TO BE ENCRYPTED
S1406
GENERATE AND ADDITIONALLY USE "Kc(0)" AND "Kc(1)" AS KEYS TO BE ENCRYPTED
S1407
IS CONTENT TYPE FIELD "2" OR "3"?
YES
NO
S1408
OBTAIN "Kc_Encrypted" AND "Kicv_cont_Encrypted" BY USING "Kdist" TO ENCRYPT DATA (CBC MODE)
S1409
OBTAIN "Kc_Encrypted" AND "Kicv_cont"_Encrypted" BY USING "Ksto" OF MEDIA 2 TO ENCRYPT DATA (CBC MODE)
END

FIG. 44

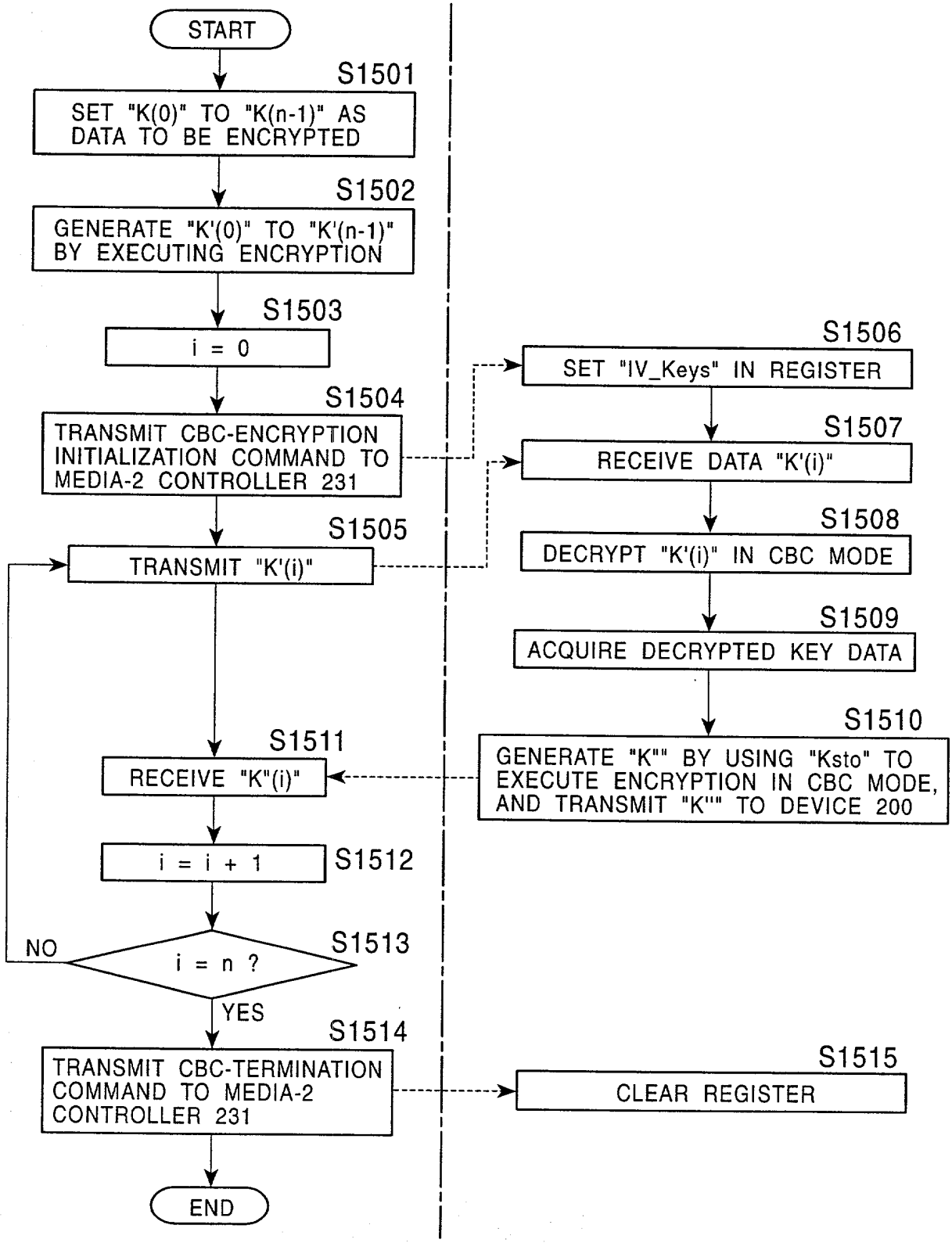

START
S1501
SET "K(0)" TO "K(n-1)" AS DATA TO BE ENCRYPTED
S1502
GENERATE "K'(0)" TO "K'(n-1)" BY EXECUTING ENCRYPTION
S1503
i = 0
S1504
TRANSMIT CBC-ENCRYPTION INITIALIZATION COMMAND TO MEDIA-2 CONTROLLER 231
S1505
TRANSMIT "K'(i)"
S1506
SET "IV_Keys" IN REGISTER
S1507
RECEIVE DATA "K'(i)"
S1508
DECRYPT "K'(i)" IN CBC MODE
S1509
ACQUIRE DECRYPTED KEY DATA
S1510
GENERATE "K"" BY USING "Ksto" TO EXECUTE ENCRYPTION IN CBC MODE, AND TRANSMIT "K"" TO DEVICE 200
S1511
RECEIVE "K"(i)"
S1512
i = i + 1
S1513
i = n ?
NO
YES
S1514
TRANSMIT CBC-TERMINATION COMMAND TO MEDIA-2 CONTROLLER 231
S1515
CLEAR REGISTER
END

FIG. 45

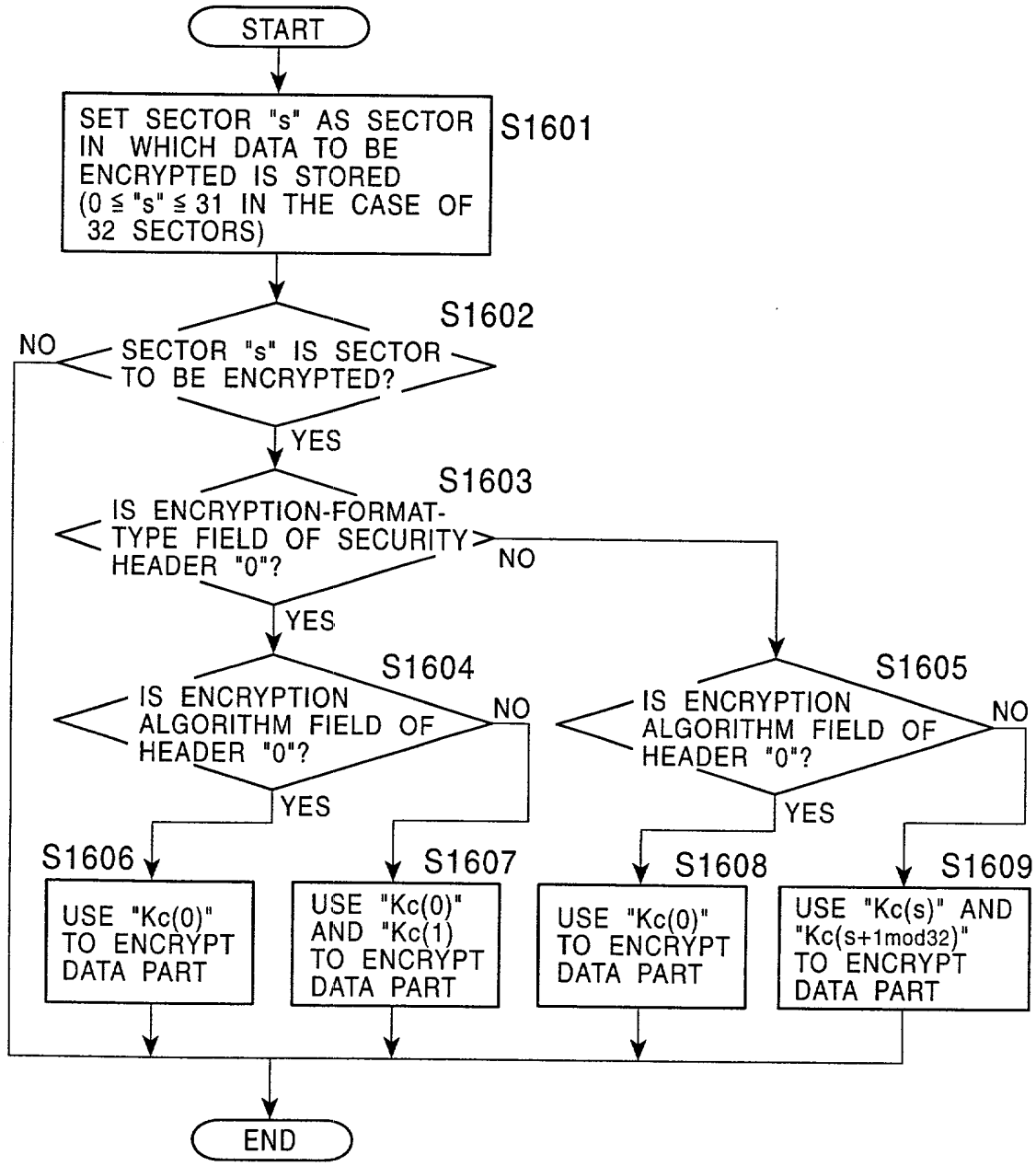

START
SET SECTOR "s" AS SECTOR IN WHICH DATA TO BE ENCRYPTED IS STORED (0 ≦ "s" ≦ 31 IN THE CASE OF 32 SECTORS)
S1601
S1602
SECTOR "s" IS SECTOR TO BE ENCRYPTED?
NO
YES
S1603
IS ENCRYPTION-FORMAT-TYPE FIELD OF SECURITY HEADER "0"?
NO
YES
S1604
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1605
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1606
USE "Kc(0)" TO ENCRYPT DATA PART
S1607
USE "Kc(0)" AND "Kc(1) TO ENCRYPT DATA PART
S1608
USE "Kc(0)" TO ENCRYPT DATA PART
S1609
USE "Kc(s)" AND "Kc(s+1mod32)" TO ENCRYPT DATA PART
END

FIG. 46

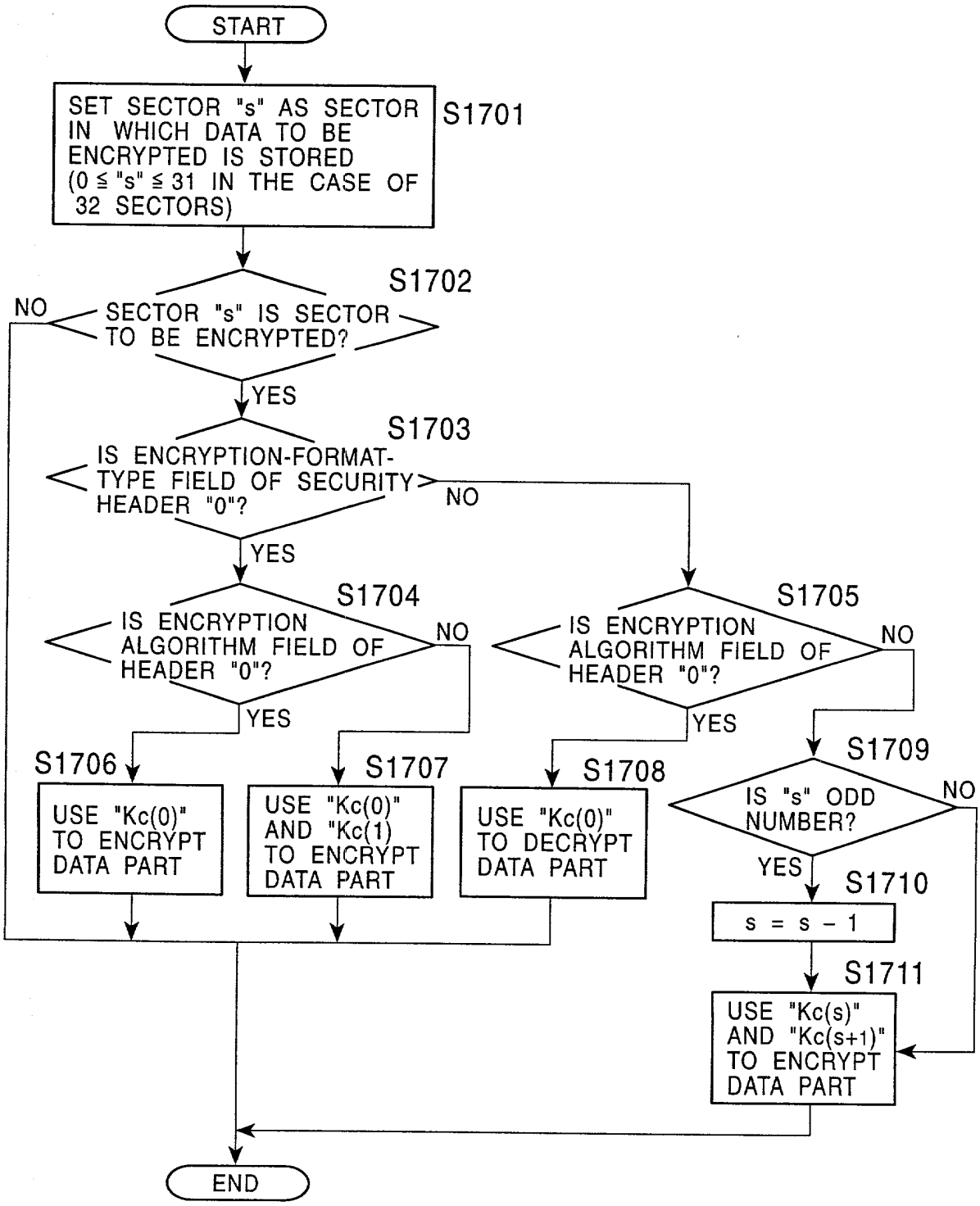
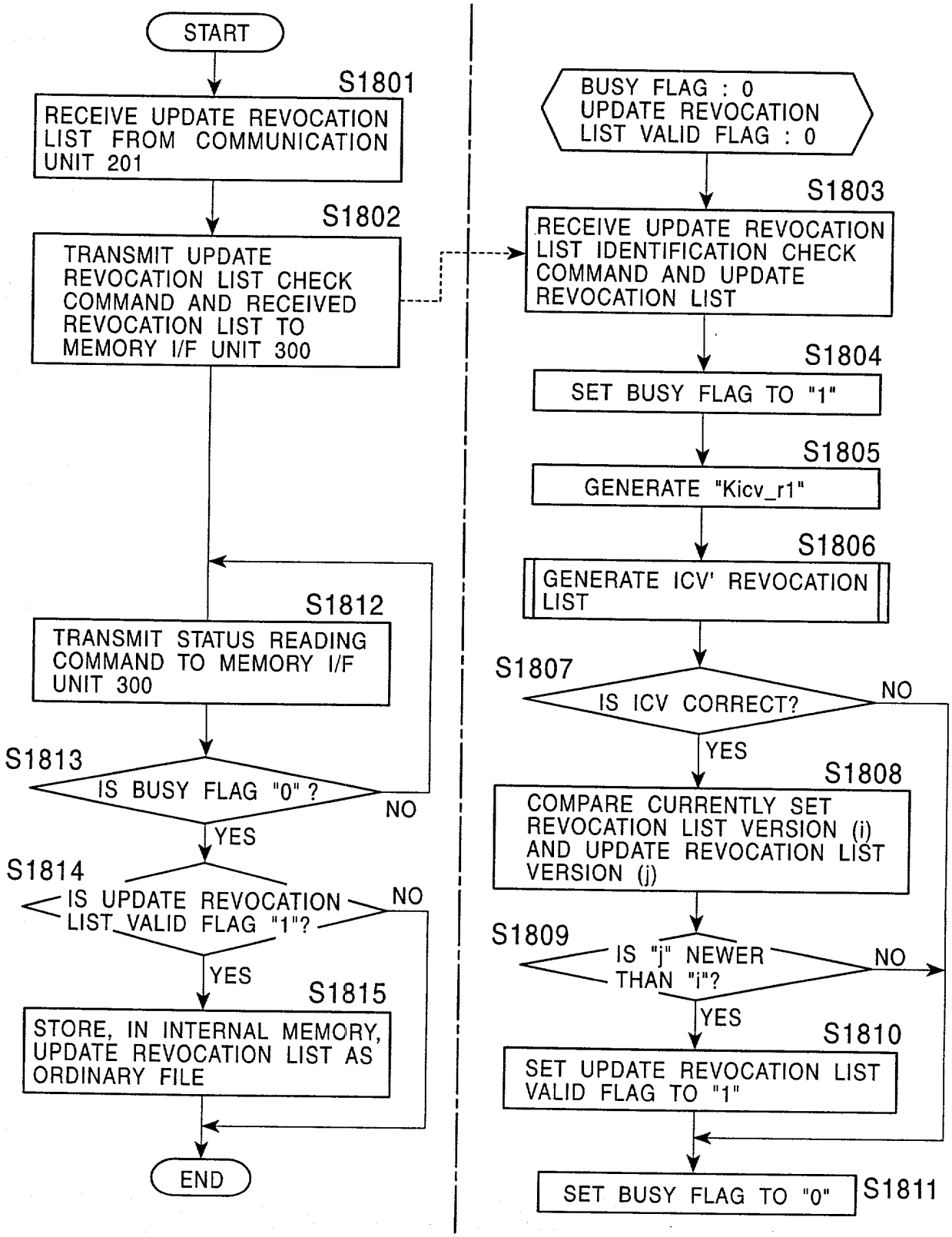

START
SET SECTOR "s" AS SECTOR IN WHICH DATA TO BE ENCRYPTED IS STORED (0 ≦ "s" ≦ 31 IN THE CASE OF 32 SECTORS)
S1701
S1702
SECTOR "s" IS SECTOR TO BE ENCRYPTED?
NO
YES
S1703
IS ENCRYPTION-FORMAT-TYPE FIELD OF SECURITY HEADER "0"?
NO
YES
S1704
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1705
IS ENCRYPTION ALGORITHM FIELD OF HEADER "0"?
NO
YES
S1706
USE "Kc(0)" TO ENCRYPT DATA PART
S1707
USE "Kc(0)" AND "Kc(1) TO ENCRYPT DATA PART
S1708
USE "Kc(0)" TO DECRYPT DATA PART
S1709
IS "s" ODD NUMBER?
NO
YES
S1710
s = s - 1
S1711
USE "Kc(s)" AND "Kc(s+1)" TO ENCRYPT DATA PART
END

FIG. 47


START
S1801
RECEIVE UPDATE REVOCATION LIST FROM COMMUNICATION UNIT 201
S1802
TRANSMIT UPDATE REVOCATION LIST CHECK COMMAND AND RECEIVED REVOCATION LIST TO MEMORY I/F UNIT 300
S1812
TRANSMIT STATUS READING COMMAND TO MEMORY I/F UNIT 300
S1813
IS BUSY FLAG "0" ?
NO
YES
S1814
IS UPDATE REVOCATION LIST VALID FLAG "1"?
NO
YES
S1815
STORE, IN INTERNAL MEMORY, UPDATE REVOCATION LIST AS ORDINARY FILE
END
BUSY FLAG : 0
UPDATE REVOCATION LIST VALID FLAG : 0
S1803
RECEIVE UPDATE REVOCATION LIST IDENTIFICATION CHECK COMMAND AND UPDATE REVOCATION LIST
S1804
SET BUSY FLAG TO "1"
S1805
GENERATE "Kicv_r1"
S1806
GENERATE ICV' REVOCATION LIST
S1807
IS ICV CORRECT?
NO
YES
S1808
COMPARE CURRENTLY SET REVOCATION LIST VERSION (i) AND UPDATE REVOCATION LIST VERSION (j)
S1809
IS "j" NEWER THAN "i"?
NO
YES
S1810
SET UPDATE REVOCATION LIST VALID FLAG TO "1"
SET BUSY FLAG TO "0"
S1811